



**TUGAS AKHIR - KI141502**

**RANCANG BANGUN SISTEM *ENTERPRISE  
RESOURCE PLANNING* PADA MODUL  
PRODUKSI YANG BERORIENTASIKAN  
*MULTITENANCY* DENGAN *DISTRIBUTED  
DATABASE***

Akbar Anugerah Syahdamar  
NRP 5112100138

Dosen Pembimbing  
Dwi Sunaryono, S.Kom, M.Kom.  
Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.

JURUSAN TEKNIK INFORMATIKA  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016

*[Halaman ini sengaja dikosongkan]*



**FINAL PROJECT - KI141502**

# **DESIGN AND IMPLEMENTATION OF PRODUCTION MODULE ON ENTERPRISE RESOURCE PLANNING WITH MULTITENANCY ORIENTED USING DISTRIBUTED DATABASE**

**AKBAR ANUGERAH SYAHDAMAR**  
**NRP 5112100138**

Supervisor  
Dwi Sunaryono, S.Kom, M.Kom.  
Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.

DEPARTMENT OF INFORMATICS  
Faculty of Information Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016

*[Halaman ini sengaja dikosongkan]*

## LEMBAR PENGESAHAN

# RANCANG BANGUN SISTEM *ENTERPRISE RESOURCE PLANNING* PADA MODUL PRODUKSI YANG BERORIENTASIKAN *MULTITENANCY* DENGAN *DISTRIBUTED DATABASE*

## TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada

Bidang Studi Manajemen Informasi  
Program Studi S-1 Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh :

**AKBAR ANUGERAH SYAHDAMAR**

NRP : 5112 100 138

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Dwi Sunaryono, S.Kom, M.Kom

NIP: 19720528 199702 1001

(pembimbing 1)

Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.

NIP: 19590803 198601 1001

(pembimbing 2)

**SURABAYA**

**JULI, 2016**

*[Halaman ini sengaja dikosongkan]*

# **RANCANG BANGUN SISTEM *ENTERPRISE RESOURCE PLANNING* PADA MODUL PRODUKSI YANG BERORIENTASIKAN *MULTITENANCY* DENGAN *DISTRIBUTED DATABASE***

Nama Mahasiswa : Akbar Anugerah Syahdamar  
NRP : 5112 100 138  
Jurusan : Teknik Informatika FTIf-ITS  
Dosen Pembimbing 1 : Dwi Sunaryono, S.Kom, M.Kom.  
Dosen Pembimbing 2 : Prof. Drs. Ec. Ir. Rivanarto Sarno,  
M.Sc.,Ph.D.

## **ABSTRAKSI**

*Saat ini fitur penjadwalan produksi di dalam aplikasi Enterprise Resource Planning (ERP) yang telah ada umumnya belum mampu untuk melakukan pemilihan sumberdaya produksi dan manajemen waktu dengan optimal. Contohnya pada Oodo ERP tidak ada Routing yang berfungsi sebagai penentu tahapan-tahapan operasi pada penjadwalan produksi. Sedangkan penjadwalan produksi pada Adempiere dan Ino ERP masih dilakukan secara manual dalam pemilihan sumberdaya dan waktu produksi. Pada ERP hasil pengembangan Tugas Akhir Teknik Informatika ITS sebelumnya ternyata juga tidak didapati fitur penjadwalan produksi. Hal ini menyebabkan perhitungan biaya produksi menjadi tidak optimal.*

*Metode penjadwalan produksi yang akan diajukan menggunakan Algoritma Genetika/Genetic Algorithm (GA). GA digunakan untuk memilih sumberdaya manusia dan/atau mesin serta waktu penjadwalan yang paling optimal. GA menerapkan konsep seleksi alam untuk mendapatkan individu/solusi terbaik dari suatu populasi menggunakan parameter-parameter genetika tertentu. Parameter genetika yang digunakan adalah kromosom,*

*nilai fitness, pindah silang, dan mutasi. Tingkat ke-optimal-an dari setiap solusi yang dihasilkan oleh GA akan diukur berdasarkan nilai fitness. Nilai fitness akan semakin besar jika rata-rata waktu produksi semakin kecil. Selain fitur penjadwalan, modul produksi juga didukung oleh beberapa submodul. Submodul tersebut adalah Routings, Bill of Material, Production Request, Production Order, Confirmation, Goods Receive, dan Goods Transfer.*

*Penerapan GA dalam penyelesaian permasalahan penjadwalan produksi menghasilkan solusi penjadwalan yang optimal. Percobaan yang dilakukan menunjukkan rata-rata waktu produksi yang semakin kecil seiring pergantian generasi. Hal tersebut secara otomatis meminimalkan biaya produksi seperti perawatan dan depresiasi mesin.*

**Kata kunci:** *Enterprise Resource Planning, ERP, Modul Produksi, Scheduling, Penjadwalan, Genetic Algorithm, GA.*



# **DESIGN AND IMPLEMENTATION OF PRODUCTION MODULE ON ENTERPRISE RESOURCE PLANNING WITH MULTITENANCY ORIENTED USING DISTRIBUTED DATABASE**

Student Name : Akbar Anugerah Syahdamar  
Student ID : 5112 100 138  
Major : Informatics Department FTIf-ITS  
Advisor 1 : Dwi Sunaryono, S.Kom, M.Kom.  
Advisor 2 : Prof. Drs. Ec. Ir. Rryanarto Sarno, M.Sc.,Ph.D.

## **ABSTRACTION**

*At this time, most of Enterprise Resource Planning (ERP) softwares do not have any optimized production scheduling. As an example, Oodo ERP does not has Routing which manage the sequence of production steps. In the other side Adempiere and Ino ERP which already have Routing but they have to manually scheduled. The latest ERP of final project also does not has any production scheduling ability. Manual adjustment of production scheduling may lead to ineffectiveness calculation of production cost.*

*The proposed method to solve production scheduling problem is Genetic Algorithm (GA). GA is proposed to find the optimal solution of production resource selection and time allocation. GA adapts the law of natural selection to find the most fit individual based on its fitness value. Fitness value in production scheduling is defined as the average time of jobs. The smaller average time means higher fitness value. In order to perform selection, GA consisted of chromosome, fitness value, cross over, and mutation. Besides that, there are some submodules which support the production scheduling. They are*

*Routings, Bill of Material, Production Request, Production Order, Confirmation, Goods Receive, and Goods Transfer.*

*The implementation of GA to solve production scheduling problem in ERP gives an optimal solution. It effectively choose the optimal resources of production and allocate production time efficiently. Therefore, the production cost can be calculated accurately.*

**Keywords:** *Enterprise Resource Planning, ERP, Production Module, Scheduling, Genetic Algorithm, GA.*

## **KATA PENGANTAR**

Puji syukur kepada Tuhan Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul:

### **RANCANG BANGUN SISTEM *ENTERPRISE RESOURCE PLANNING* PADA MODUL PRODUKSI YANG BERORIENTASIKAN *MULTITENANCY* DENGAN *DISTRIBUTED DATABASE***

Melalui lembar ini, penulis ingin menyampaikan ucapan terimakasih dan penghormatan yang sebesar-besarnya kepada:

1. Bapak, Ibu, kakak dan keluarga besar yang selalu memberikan dukungan penuh untuk menyelesaikan Tugas Akhir ini.
2. Bapak Riyanarto Sarno beserta keluarga dan Bapak Dwi Sunaryono beserta keluarga selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan petunjuk selama proses pengerjaan tugas akhir ini.
3. Bapak dan Ibu dosen Jurusan Teknik Informatika ITS yang telah banyak menyampaikan ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
4. Seluruh Staf dan karyawan FTIf ITS yang banyak memberikan banyak kelancaran administrasi akademik kepada penulis.
5. Al Aji, Vicky, Arya, Ashari, Nicko, Yusi, Melinda dan Afina yang telah menemani dan membantu pengerjaan Tugas Akhir ini.
6. Teman-teman Laboratorium Manajemen Informasi sekaligus teman-teman seperjuangan anak didik Tugas Akhir Prof. Riyanarto yaitu Al Aji, Vicky, Arya, Ashari, Nicko, Yusi, Melinda, Afina.

7. Teman-teman Administrator Laboratorium Manajemen Informasi yaitu Naufal Fakhri, Riska Adhita, Miftah, Kania, Anne, Nadia, Nanang dan Lino.
8. Teman-teman angkatan 2012 jurusan Teknik Informatika ITS yang telah menemani dan mendukung penulis selama kuliah.
9. Bapak Mujib sebagai penjaga parkir jurusan Teknik Informatika yang telah menjaga sepeda motor penulis selama pengerjaan Tugas Akhir ini.
10. Serta pihak-pihak lain yang namanya tidak dapat penulis sebutkan satu per satu.

Bagaimanapun juga penulis telah berusaha sebaik-baiknya dalam menyusun Tugas Akhir ini, namun penulis mohon maaf apabila terdapat kekurangan yang penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Juni 2016

Akbar Anugerah Syahdamar

## DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAKSI.....	vii
ABSTRACTION.....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xix
DAFTAR TABEL.....	xxiii
DAFTAR KODE SUMBER.....	xxv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Permasalahan.....	3
1.3. Batasan Permasalahan.....	3
1.4. Tujuan.....	4
1.5. Manfaat.....	4
1.6. Metodologi.....	4
1.7. Sistematika Penulisan.....	5
BAB II DASAR TEORI.....	7
.....	7
2.1. Penelitian Terkait.....	7
2.2. ERP ( <i>Enterprise Resource Planning</i> ).....	9
2.3. Modul Produksi.....	9
2.3.1. <i>Bill of Material</i> .....	12
2.3.2. <i>Routings</i> .....	12
2.3.3. <i>Request for Production</i> .....	12
2.3.4. <i>Production Order</i> .....	12
2.3.5. <i>Goods Receive</i> .....	13
2.3.6. <i>Confirmation</i> .....	13
2.3.7. <i>Goods Transfer</i> .....	13
2.4. Algoritma Genetika.....	13

2.4.1. Individu/Kromosom.....	14
2.4.2. Persilangan/ <i>Crossover</i> .....	17
2.4.3. <i>Fitness Function</i> .....	18
2.4.4. Mutasi.....	18
2.5. <i>Multitenant Data Architecture</i> atau <i>Multitenancy</i> .....	18
2.5.1. Separated Database.....	19
2.5.2. Shared Database, Separated Schema.....	19
2.5.3. Shared Database, Shared Schema.....	20
2.6. <i>Distributed Database</i> .....	21
2.6.1. Replikasi.....	21
2.6.2. Fragmentasi.....	21
2.7. MySQL Cluster.....	21
2.8. PHP (Hypertext Preprocessor).....	22
2.9. Framework Yii 2.....	22
BAB III ANALISIS DAN PERANCANGAN SISTEM.....	23
.....	23
3.1. Analisis.....	23
3.1.1. Analisis Proses Bisnis.....	23
3.1.2. Analisis Data.....	32
3.2. Deskripsi Umum Sistem.....	32
3.3. Aktor.....	33
3.4. Spesifikasi Kebutuhan Perangkat Lunak.....	34
3.5. Physical Data Model.....	35
3.6. Kasus Penggunaan.....	39
3.6.1. Analisis Kasus Penggunaan Mengelola <i>Routings</i> .....	40
3.6.2. Analisis Kasus Penggunaan Mengelola Operasi.....	43
3.6.3. Analisis Kasus Penggunaan Mengelola <i>Bill of Material</i> .....	45
3.6.4. Analisis Kasus Penggunaan Membuat <i>Request for Production</i> .....	47

3.6.5. Analisis Kasus Melakukan <i>Confirmation</i> .....	48
3.6.6. Analisis Kasus Penggunaan Membuat <i>Goods Receive</i> .....	50
3.6.7. Analisis Kasus Penggunaan Membuat <i>Goods Transfer</i> .....	52
3.6.8. Analisis Kasus Penggunaan <i>Generating Scheduling</i> .....	54
3.7. Perancangan <i>Scheduling</i> .....	56
3.7.1. Pengkodean Kromosom.....	57
3.7.2. Pindah Silang.....	60
3.7.3. Mutasi.....	61
3.7.4. Fitness Function.....	61
3.8. Perancangan Basis Data Terdistribusi.....	62
3.9. Perancangan Multitenancy.....	62
3.10. Perancangan RBAC.....	63
3.10.1. Perancangan Antarmuka <i>Login</i> .....	63
3.10.2. Perancangan Antarmuka <i>Add User</i> .....	64
3.11. Perancangan Antarmuka Sistem.....	65
3.11.1. Perancangan Antarmuka Halaman <i>Input Data Routings</i> .....	66
3.11.2. Perancangan Antarmuka Halaman <i>Input Data Operasi</i> .....	67
3.11.3. Perancangan Antarmuka Halaman <i>Input Data Bill of Material</i> .....	67
3.11.4. Perancangan Antarmuka Halaman <i>Input Data Request for Production</i> .....	67
3.11.5. Perancangan Antarmuka Halaman <i>Goods Receive</i> .....	68
3.11.6. Perancangan Antarmuka Halaman <i>Confirmation</i> .....	68
3.11.7. Perancangan Antarmuka Halaman <i>Goods Transfer</i> .....	69
3.11.8. Perancangan Antarmuka Halaman <i>Generating Scheduling</i> .....	69

BAB IV IMPLEMENTASI SISTEM.....	71
.....	71
4.1. Lingkungan Pengembangan Sistem.....	71
4.2. Implementasi Basis Data Terdistribusi.....	71
4.2.1. Instalasi Data dan SQL node pada node1 dan node2	71
4.2.2. Pemasangan Node Management pada node03.....	73
4.2.3. Konfigurasi Manajemen Node.....	74
4.2.4. Konfigurasi Data dan SQL Node.....	75
4.2.5. Memulai MySQL Cluster.....	76
4.3. Implementasi RBAC (singkatan).....	78
4.3.1. Membuat Tabel Pengguna.....	78
4.3.2. Membuat 4 Tabel Autentifikasi RBAC dan Tabel Pengguna.....	79
4.3.3. Membuat Modul Admin.....	80
4.3.4. Membuat Model Tabel Autentifikasi, <i>Controller</i> dan <i>View</i> Pengguna.....	80
4.3.5. Menambahkan Kode pada user controller modul admin.....	81
4.4. Implementasi Multitenancy.....	81
4.4.1. Membuat Halaman Muka Tenant.....	81
4.4.2. Menambah <i>Database</i> untuk Tenant Baru.....	82
4.4.3. <i>Login</i> Tenant.....	82
4.5. Implementasi Antarmuka Modul Produksi.....	83
4.5.1. Implementasi Antarmuka Formulir <i>Input</i> data <i>Routings</i> .....	84
4.5.2. Implementasi Antarmuka Formulir <i>Input</i> data Operasi	84
4.5.3. Implementasi Antarmuka Formulir <i>Input</i> data <i>Bill of Material</i> .....	85



4.5.4. Implementasi Antarmuka Formulir <i>Input data Request for Production</i> .....	86
4.5.5. Implementasi Antarmuka Formulir <i>Confirmation data Production Order</i> .....	87
4.5.6. Implementas Antarmuka Formulir <i>Input data Goods Receive</i> .....	88
4.5.7. Implementasi Antarmuka Formulir <i>Input data Goods Transfer</i> .....	89
4.5.8. Implementasi Antarmuka <i>Generating Scheduling</i> ....	90
4.6. Implementasi Program Modul Produksi.....	91
4.6.1. Program <i>Routing</i> .....	91
4.6.2. Program Operasi.....	92
4.6.3. Program <i>Bill of Material</i> .....	92
4.6.4. Program <i>Request for Production</i> .....	93
4.6.5. Program <i>Confirmation</i> .....	93
4.6.6. Program <i>Goods Receive</i> .....	94
4.6.7. Program <i>Goods Transfer</i> .....	95
4.7. Implementasi Program <i>Scheduling</i> .....	96
4.7.1. Pengkodean Kromosom.....	96
4.7.2. Pindah Silang.....	97
4.7.3. Mutasi.....	99
4.7.4. <i>Fitness Function</i> .....	99
BAB V PENGUJIAN DAN EVALUASI.....	101
.....	101
5.1. Lingkungan Pengujian.....	101
5.2. Skenario Pengujian Fungsionalitas Modul Produksi.....	101
5.2.1. Pengujian Fitur Mengelola data <i>Routing</i> .....	101
5.2.2. Pengujian Fitur Mengelola Data Operasi.....	105

5.2.3. Pengujian Fitur Mengelola data <i>Bill of Material</i> .....	109
5.2.4. Pengujian Fitur Membuat <i>Request for Production</i> ..	113
5.2.5. Pengujian Fitur <i>Confirmation</i> .....	114
5.2.6. Pengujian Fitur Menambah <i>Goods Receive</i> .....	119
5.2.7. Pengujian Fitur Menambah <i>Goods Transfer</i> .....	122
5.2.8. Pengujian Fitur <i>Generating Scheduling</i> .....	124
5.3. Pengujian Fitur Basis Data Terdistribusi.....	128
5.3.1. Replikasi.....	128
5.3.2. <i>High Availability</i> .....	130
5.4. Evaluasi Pengujian Fungsionalitas.....	139
BAB VI KESIMPULAN DAN SARAN.....	141
.....	141
6.1. Kesimpulan.....	141
6.2. Saran.....	141
DAFTAR PUSTAKA.....	143
LAMPIRAN A - RINCIAN PHYSICAL DATA MODEL.....	145
A.1Daftar <i>Table</i> pada PDM.....	145
DAFTAR ISTILAH.....	153
INDEKS.....	155
BIODATA PENULIS.....	157

## DAFTAR GAMBAR

Gambar 2.1	Keterkaitan Antara Modul Produksi dan <i>Inventory and Warehouse Management</i> .....	10
Gambar 2.2	Proses Bisnis <i>Confirmation</i> .....	11
Gambar 2.3	Susunan Kromosom.....	14
Gambar 2.4	Contoh Kromosom.....	15
Gambar 2.5	Proses Persilangan.....	17
Gambar 2.6	<i>Separated Schema, Separated Database</i> .....	19
Gambar 2.7	<i>Shared Database, Separated Schema</i> .....	20
Gambar 2.8	<i>Shared Schema, Shared Database</i> .....	20
Gambar 3.1	Gambaran Umum Proses Bisnis ERP.....	24
Gambar 3.2	Skema <i>Make to Order</i> .....	25
Gambar 3.3	Skema <i>Make to Stock</i> .....	26
Gambar 3.4	Penjabaran <i>Creating Request for Production</i> .....	27
Gambar 3.5	Penjabaran <i>Authorizing Production</i> .....	28
Gambar 3.6	Penjabaran <i>Releasing Production Order</i> .....	29
Gambar 3.7	Penjabaran <i>Receiving Goods</i> .....	30
Gambar 3.8	Penjabaran <i>Transferring Goods</i> .....	31
Gambar 3.9	Gambaran Umum Sistem.....	33
Gambar 3.10	<i>Physical Data Model (PDM)</i> .....	38
Gambar 3.11	Diagram Kasus Penggunaan.....	39
Gambar 3.12	Diagram Aktivitas Mengelola <i>Routings</i> .....	42
Gambar 3.13	Diagram Aktivitas Mengelola Operasi.....	44
Gambar 3.14	Diagram Aktivitas Mengelola <i>Bill of Material</i> .....	46
Gambar 3.15	Diagram Aktivitas Membuat <i>Request for Production</i> .....	48
Gambar 3.16	Diagram Aktivitas Melakukan <i>Confirmation</i> .....	50
Gambar 3.17	Diagram Aktivitas Membuat <i>Goods Receive</i> .....	52
Gambar 3.18	Diagram Aktivitas Membuat <i>Goods Transfer</i> .....	54
Gambar 3.19	Diagram Aktivitas <i>Generating Scheduling</i> .....	56
Gambar 3.20	Perancangan <i>Multi-tenancy</i> .....	63
Gambar 3.21	Antarmuka <i>Login</i> .....	64
Gambar 3.22	Antarmuka <i>Add User</i> .....	64
Gambar 3.23	<i>Template</i> Perancangan Antarmuka Sederhana .....	65

Gambar 3.24 <i>Template</i> Perancangan Antarmuka Rumit.....	66
Gambar 3.25 <i>Template</i> Perancangan Antarmuka <i>list</i> .....	66
Gambar 4.1 Berhasil Menjalankan <i>Management Node</i> .....	76
Gambar 4.2 Ndbd Berhasil Dijalankan.....	77
Gambar 4.3 MySQL Telah Aktif.....	77
Gambar 4.4 Data Node Saling Terkoneksi.....	78
Gambar 4.5 Proses Migrasi Tabel Pengguna Berhasil.....	79
Gambar 4.6 Proses Migrasi Tabel Pengguna Berhasil.....	80
Gambar 4.7 Implementasi Antarmuka <i>Input Data Routings</i> .....	84
Gambar 4.8 Implementasi Antarmuka <i>Input Data Operasi</i> .....	85
Gambar 4.9 Implementasi Antarmuka <i>Input Data Bill of Material</i> .....	86
Gambar 4.10 Implementasi Antarmuka <i>Input Data Request for Production</i> .....	87
Gambar 4.11 Implementasi Antarmuka Formulir <i>Confirmation</i> .....	88
Gambar 4.12 Implementasi Antarmuka <i>Input Data Goods Receive</i> .....	89
Gambar 4.13 Implementasi Antarmuka <i>Input Data Goods Transfer</i> .....	90
Gambar 4.14 Implementasi Antarmuka <i>Input Data Goods Receive</i> .....	91
Gambar 5.1 Pengujian Penambahan Data <i>Routings Tire for All Bike</i> .....	104
Gambar 5.2 Penyuntingan Data <i>Routings Tire for All Bike</i> .....	104
Gambar 5.3 Data Sebelum Penghapusan <i>Routings Tire for All Bike</i> .....	104
Gambar 5.4 Setelah Penghapusan Data <i>Routings Tire for All Bike</i> .....	105
Gambar 5.5 Melihat Data <i>Routings</i> .....	105
Gambar 5.6 Penambahan Data Operasi.....	107
Gambar 5.7 Sebelum Perubahan Data Operasi <i>Checking Material</i> .....	108
Gambar 5.8 Setelah Perubahan Data Operasi <i>Checking Material</i> .....	108
Gambar 5.9 Sebelum Penghapusan Data Operasi ID 21.....	108
Gambar 5.10 Setelah Penghapusan Data Operasi ID 21.....	108
Gambar 5.11 Melihat Data Operasi ID 1.....	109
Gambar 5.12 Penambahan Data <i>BOM Tire for All Bike</i> .....	111
Gambar 5.13 Setelah Perubahan Data <i>BOM Tire for All Bike</i> .....	112

Gambar 5.14	Sebelum Penghapusan Data <i>BOM Tire for All Bike</i>	112
Gambar 5.15	Setelah Penghapusan Data <i>BOM Tire for All Bike</i>	112
Gambar 5.16	Melihat Data <i>BOM ID 8</i>	113
Gambar 5.17	Penambahan Data <i>Request for Production : Deluxe Touring Bike with Red Main Color</i>	114
Gambar 5.18	Data <i>Confirmation</i> 100 buah <i>Professional Bike with Black Color</i>	115
Gambar 5.19	Daftar Mesin yang Terlibat Pada Proses Produksi	116
Gambar 5.20	<i>Usage</i> Setiap Mesin Sebelum <i>Confirmation</i>	118
Gambar 5.21	<i>Usage</i> Setiap Mesin Sesudah <i>Confirmation</i>	119
Gambar 5.22	Data <i>Goods Receive ID 82</i>	121
Gambar 5.23	Data <i>Goods Issue ID 94</i>	122
Gambar 5.24	Data <i>Goods Transfer 1</i>	123
Gambar 5.25	Data <i>Goods Receipt 1</i>	124
Gambar 5.26	Data <i>Routings : Deluxe Touring Bike with Black Main Color</i>	126
Gambar 5.27	Data Sumberdaya Manusia	126
Gambar 5.28	Data Sumberdaya Mesin	127
Gambar 5.29	Pengujian fitur Replikasi pada sistem	129
Gambar 5.30	Pengujian Fitur Replikasi pada Database Server	129
Gambar 5.31	Pengujian Fitur Replikasi pada Database Server	2130
Gambar 5.32	Pengujian Fitur High-Availability pada Sistem	131
Gambar 5.33	Pengujian Fitur High-Availability pada Database Server 2	132
Gambar 5.34	Pengujian Fitur High-Availability pada Database Server 1	133
Gambar 5.35	Hasil Pengujian <i>Multitenancy</i>	135
Gambar 5.36	Proses login admin	137
Gambar 5.37	Tampilan awal setelah login berhasil dilakukan	137
Gambar 5.38	Proses admin menambahkan user baru	138
Gambar 5.39	Proses menyunting data user oleh admin	138
Gambar 5.40	Proses penghapusan user oleh admin	139
Gambar A.1	<i>Table</i> Barang diterima	145
Gambar A.2	<i>Table Bill of Material</i>	145
Gambar A.3	<i>Table</i> Bom disusun oleh	146

Gambar A.4	<i>Table Goods Transfer</i> .....	146
Gambar A.5	<i>Table Goods Receive</i> .....	147
Gambar A.6	<i>Table Item</i> .....	147
Gambar A.7	<i>Table Job Order</i> .....	148
Gambar A.8	<i>Table Mesin</i> .....	148
Gambar A.9	<i>Table Operasi</i> .....	149
Gambar A.10	<i>Table Pegawai</i> .....	149
Gambar A.11	<i>Table Production Order</i> .....	150
Gambar A.12	<i>Table Request for Production</i> .....	150
Gambar A.13	<i>Table Routings</i> .....	151
Gambar A.14	<i>Table Routings disusun oleh</i> .....	151

## DAFTAR TABEL

Tabel 2.1 Perbedaan Modul ERP 2013 dengan 2016.....	7
Tabel 2.2 Perbandingan ERP 2013 dan 2016 pada Modul Produksi.....	8
Tabel 2.3 Keterkaitan Antar Modul.....	11
Tabel 2.4 Sumberdaya per Operasi.....	14
Tabel 2.5 Contoh Alokasi Sumberdaya per Operasi.....	15
Tabel 3.1 Kebutuhan Fungsional Sistem.....	34
Tabel 3.2 Keterangan Tabel pada PDM.....	35
Tabel 3.3 Tabel Kode Kasus Penggunaan.....	39
Tabel 5.1 Daftar Mesin yang Terlibat.....	116
Tabel 5.2 Mesin yang Telah Digunakan Berdasar <i>Query</i> .....	116
Tabel 5.3 <i>Usage</i> Mesin Disaat Sebelum dan Sesudah <i>Confirmation</i> .....	117
Tabel 5.4 Optimasi Penjadwalan Per Iterasi.....	127
Tabel 5.5 Rangkuman Hasil Pengujian.....	139

*[Halaman ini sengaja dikosongkan]*



## DAFTAR KODE SUMBER

Kode Sumber 3.1 Pseudocode Membuat Himpunan Operasi Awal	57
Kode Sumber 3.3 Pseudocode Mencari Operasi dengan Sumberdaya yang Sama.....	58
Kode Sumber 3.4 Pseudocode Mencari Operasi Paling Awal.....	58
Kode Sumber 3.5 Pseudocode Hapus Operasi Tercepat B dari A	59
Kode Sumber 3.6 Pseudocode Memperbarui Waktu Operasi dari <i>Jobs</i> yang Terpilih dari B.....	59
Kode Sumber 3.7 Pseudocode Memasukan Operasi Selanjutnya	60
Kode Sumber 3.8 Pseudocode Ulangi Semua Langkah Hingga A Habis.....	60
Kode Sumber 3.9 Pseudocode Pindah Silang.....	61
Kode Sumber 3.10 Pseudocode Perancangan Mutasi Algoritma Genetika.....	61
Kode Sumber 3.11 Pseudocode <i>Fitness Function</i> .....	62
Kode Sumber 4.1 Membuat grup MySQL pengguna baru dan menambah user MySQL.....	72
Kode Sumber 4.2 Mengubah lokasi ke dalam direktori.....	72
Kode Sumber 4.3 Menjalankan <i>script</i> MySQL.....	72
Kode Sumber 4.4 Mengatur izin yang diperlukan oleh server MSQl.....	73
Kode Sumber 4.5 Menyalin script startup MySQL.....	73
Kode Sumber 4.6 Ekstraksi <i>ndb_mgm</i> dan <i>ndb_mgmd</i> .....	74
Kode Sumber 4.7 Mengubah Lokasi Direktori.....	74
Kode Sumber 4.8 Membuat Direktori Tempat File Konfigurasi.	74
Kode Sumber 4.9 Mengatur File “ <i>config.ini</i> ”.....	75
Kode Sumber 4.10 Konfigurasi SQL <i>Node</i> .....	75
Kode Sumber 4.11 Memulai Manajemen <i>Node</i> .....	76
Kode Sumber 4.12 Memulai <i>ndbd</i> .....	76
Kode Sumber 4.13 Memulai MySQL.....	77
Kode Sumber 4.14 <i>Generate</i> tabel RBAC.....	79
Kode Sumber 4.15 Fungsi Tabel <i>AuthItem</i> .....	81
Kode Sumber 4.16 Pembuatan Halaman Muka Tenant.....	82

Kode Sumber 4.19 <i>Method</i> Membuat Data <i>Routings</i> .....	92
Kode Sumber 4.20 <i>Method</i> Membuat Data Operasi.....	92
Kode Sumber 4.21 <i>Method</i> Membuat Data <i>Bill of Material</i> .....	93
Kode Sumber 4.22 <i>Method</i> Menambah Data <i>Request for Production</i> .....	93
Kode Sumber 4.23 <i>Method Confirmation</i> .....	94
Kode Sumber 4.24 <i>Method</i> Membuat <i>Goods Receive</i> .....	95
Kode Sumber 4.26 Pengkodean Kromosom.....	97
Kode Sumber 4.27 Pindah Silang dan <i>Roulette Wheel</i> .....	98
Kode Sumber 4.28 Implementasi Mutasi.....	99
Kode Sumber 4.29 Implementasi <i>Fitness Function</i> .....	100

## **BAB I**

### **PENDAHULUAN**

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

#### **1.1. Latar Belakang**

Kondisi dunia bisnis saat ini telah berkembang menjadi semakin kompleks, semakin kompetitif, bergerak dengan cepat serta semakin sulit untuk diprediksi. Agar dapat bersaing dan sukses, perusahaan perlu memadukan bisnis dan sumber daya IT yang dimiliki agar dapat secara fleksibel mengakomodasi adanya perubahan untuk kemudian dilakukan adaptasi terhadap perubahan tersebut secara cepat dan tepat.

Dari pembahasan diatas dapat disimpulkan bahwa kebutuhan akan aplikasi *Enterprise Resource Planning* (ERP) dalam perusahaan merupakan kebutuhan yang perlu agar perusahaan dapat memadukan bisnis dan sumber daya. Pada ERP terdapat banyak sekali modul-modul yang dibuat dan diintegrasikan, antara lain adalah *Finance / Accounting, Sales and Delivery, Inventory, Production and Planning, Human Resource Management*, dan lainnya. Pada tugas akhir ini, penulis akan membangun paket atau modul aplikasi *Production*.

Sistem ERP yang akan dibangun ini, khususnya untuk modul *Production* sebagian besar berkiblat pada aplikasi ERP nomor satu di dunia yaitu SAP, di dalam SAP, modul *Production* terdapat dalam satu paket, yaitu *SAP PP (Production and Planning)*.

Piranti lunak SAP, khususnya pada modul *Production and Planning* ditujukan untuk mempermudah manajemen produksi pada sebuah perusahaan. Seluruh hal yang berkaitan dengan produksi, seperti *Bill of Material*, *Routings*, *Job Orders*, hingga transaksi bahan baku merupakan jangkauan dari modul ini.

Dengan berkaca pada modul *SAP PP* dan dengan sedikit penyesuaian sesuai kebutuhan, maka akan dibangun Modul Produksi dengan beberapa fungsionalitas utama, yaitu *Bill of Material*, *Routings*, *Request for Production*, *Production Order and Scheduling*, *Goods Receive*, *Confirmation*, dan *Goods Transfer (Completion)*.

Di dalam sistem produksi, terdapat sebuah submodul inti yaitu *Scheduling*. *Scheduling* (penjadwalan) merupakan mekanisme pembuatan jadwal dan urutan produksi dari suatu barang. Terdapat beberapa 2 inti dari *Scheduling*, yaitu pengalokasian sumber daya (mesin dan pekerja), dan manajemen waktu. Hal ini sangat penting mengingat biaya produksi merupakan parameter utama dalam penentuan harga jual sebuah produk. Berdasarkan hal tersebut, modul produksi yang akan dikembangkan akan menggunakan *Genetic Algorithm* (GA). Algoritma ini sangat populer di dalam bidang optimasi, khususnya *scheduling*. Selain faktor kehandalan, pemilihan GA didasarkan pada performa dan kedinamisan GA.

Modul produksi yang akan dibangun akan berbasiskan *website*. Aplikasi berbasis *website* dianggap memiliki nilai kepraktisan yang tinggi karena setiap pemangku kepentingan yang terlibat dapat mengaksesnya dimanapun dan kapanpun. Modul ini juga menerapkan sistem *multi-tenancy*, yaitu mekanisme penyimpanan beberapa basis data dari beberapa *client* yang berbeda di dalam sebuah server fisik. Sehingga, data dari banyak perusahaan cukup disimpan di dalam satu *server* fisik saja. Selain itu, diterapkan juga prinsip *distributed database* (basis data terdistribusi), yaitu penempatan data di banyak lokasi, tetapi

menerapkan suatu mekanisme tertentu sehingga menjadi sebuah kesatuan basis data.

## 1.2. Rumusan Permasalahan

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana implementasi ERP untuk modul Produksi yang meliputi submodul *Bill of Material*, *Routings*, *Request for Production*, *Production Order and Scheduling*, *Goods Receive*, *Confirmation*, dan *Goods Transfer (Completion)*?
2. Bagaimana penerapan *Genetic Algorithm* (GA) pada submodul *Scheduling*?
3. Bagaimana mengimplementasikan *multi-tenancy* dan *distributed database* pada modul Produksi?

## 1.3. Batasan Permasalahan

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, di antaranya sebagai berikut:

1. Aplikasi yang dikembangkan merupakan aplikasi *web*.
2. Bahasa pemrograman yang digunakan adalah PHP.
3. Modul Produksi dengan submodul:
  - a. *Routings*,
  - b. *Bill of Material*,
  - c. *Request for Production*,
  - d. *Production Order and Scheduling*,
  - e. *Goods Receive*,
  - f. *Confirmation*, dan
  - g. *Goods Transfer*.
4. Basis Data Terdistribusi menggunakan MySQL Cluster, dengan rincian:
  - a. menggunakan mekanisme replikasi penuh, dan
  - b. Disusun oleh 2 *data node*, 1 *management node*, dan 1 *web server*.

- c. Sistem ERP yang dibangun dalam aspek *multi-tenancy* dengan mekanisme *separated database*

#### **1.4. Tujuan**

Tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Membuat aplikasi ERP pada domain modul Produksi yang dapat mengakomodasi manajemen seluruh proses produksi untuk meningkatkan efisiensi dan menekan biaya produksi dari suatu perusahaan.
2. Menerapkan *Genetic Algorithm* (GA) pada submodul *Scheduling*.
3. Mengimplementasikan *multi-tenancy* dan *distributed database* pada modul Produksi.

#### **1.5. Manfaat**

Manfaat dari hasil tugas akhir ini adalah memberikan kontribusi untuk membangun aplikasi ERP (*Enterprise Resource Planning*) yang berguna untuk kalangan UKM di Indonesia demi meningkatkan pendapatan mereka. Dengan adanya aplikasi ini, diharapkan aplikasi ERP lebih berkembang untuk penggunaan skala UKM.

#### **1.6. Metodologi**

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yaitu:

1. Studi literatur

Dalam pembuatan Tugas Akhir ini telah dipelajari tentang hal-hal yang dibutuhkan sebagai ilmu penunjang dalam penyelesaiannya. Pertama adalah tentang proses bisnis di dalam Modul Produksi.

2. Penentuan Metode

Pada tahap ini penulis menjabarkan metode yang dapat

digunakan untuk optimasi penjadwalan.

### 3. Analisis dan Perancangan Sistem

### 4. Implementasi

Pada tahap ini dilakukan pembuatan elemen perangkat lunak yang merupakan implementasi dari rancangan pada proses

### 5. Pengujian dan evaluasi

Pada tahap ini dilakukan pengujian terhadap elemen perangkat lunak dengan menggunakan data uji yang telah dipersiapkan. Pengujian dan evaluasi perangkat lunak dilakukan untuk mengevaluasi jalannya perangkat lunak, mengevaluasi fitur utama, mengevaluasi fitur-fitur tambahan, mencari kesalahan yang timbul pada saat perangkat lunak aktif, dan mengadakan perbaikan jika ada kekurangan.

### 6. Penyusunan buku Tugas Akhir

Pada tahap ini dilakukan pendokumentasian dan pelaporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan Tugas Akhir.

## 1.7. Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

### Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan

masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

## **Bab II Dasar Teori**

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan yang menjadi dasar dari pembuatan Tugas Akhir ini.

## **Bab III Analisis dan Perancangan Sistem**

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan perangkat lunak meliputi perancangan alur, proses dan perancangan antarmuka pada perangkat lunak.

## **Bab IV Implementasi**

Bab ini berisi implementasi dari perancangan perangkat lunak perangkat lunak dan implementasi fitur-fitur penunjang perangkat lunak.

## **Bab V Pengujian dan Evaluasi**

Bab ini membahas pengujian dengan metode pengujian subjektif untuk mengetahui penilaian aspek kegunaan (*usability*) dari perangkat lunak dan pengujian fungsionalitas yang dibuat dengan memperhatikan keluaran yang dihasilkan serta evaluasi terhadap fitur-fitur perangkat lunak.

## **Bab VI Kesimpulan**

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

## **Daftar Pustaka**

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.



## BAB II DASAR TEORI

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir.

### 2.1. Penelitian Terkait

Tugas akhir ERP pada tahun 2013 membahas mengenai implementasi *Service Oriented Architecture* (SOA) pada modul produksi. Pada Tabel 2.1 dijabarkan modul-modul yang ada di ERP 2013 dan ERP 2016. Modul produksi yang dikembangkan memiliki beberapa fungsionalitas utama yaitu *data master operation*, *Routings*, *Bill of Material*, dan *Production Order* [9]. Sistem yang akan dikembangkan saat ini merupakan pengembangan dari ERP sebelumnya, dimana terdapat beberapa tambahan fitur seperti yang dijelaskan pada Tabel 2.2.

**Tabel 2.1 Perbedaan Modul ERP 2013 dengan 2016**

No.	Nama Modul ERP 2013	Nama Modul ERP 2016
1	Account Payable & Receivable	Account Payable & Receivable
2	General Ledger	General Ledger
3	Cost Accounting	Finance
4	Customer Relationship Management	Sales and Distribution
5	Production Planning	<b>Production</b>
6	Supplier Relationship Management	Purchasing
7	Inventory	Inventory and Warehouse Management
8	Assets Management	Assets Management
9	-	Human Resource Management

**Tabel 2.2 Perbandingan ERP 2013 dan 2016 pada Modul Produksi**

No.	Fitur ERP 2013	Fitur yang akan dikembangkan	Penjelasan
1.	<i>Data master operation</i>	Operasi	Manajemen operasi di dalam proses produksi.
2.	<i>Routings</i>	<i>Routings</i>	Manajemen urutan operasi.
3.	<i>Bill of Material</i>	<i>Bill of Material</i>	Manajemen Komposisi produk.
4.	<i>Production Order</i>	<i>Production Order</i>	Proses produksi.
5.	-	<i>Request for Production</i>	Manajemen permintaan produksi.
6.	-	<i>Scheduling</i>	Automatisasi penjadwalan proses produksi.
7.	-	<i>Goods Receive</i>	Manajemen barang masuk dari pergudangan.
8.	-	<i>Goods Transfer</i>	Manajemen barang keluar hasil produksi.

Berdasarkan Tabel 2.2 ERP 2013 dan ERP 2016 memiliki beberapa persamaan dan perbedaan fitur khususnya pada Modul Produksi. Beberapa fitur yang sama adalah *Data master operation*/Operasi, *Routings*, *Bill of Material*, dan *Production Order*. Namun, pada Modul Produksi ERP 2016 terdapat beberapa fitur tambahan seperti *Request for Production*, *Scheduling*, *Goods Receive*, dan *Goods Transfer*. Hal ini merupakan kelebihan ERP 2016 pada sisi kelengkapan fungsionalitas.

## 2.2. ERP (*Enterprise Resource Planning*)

ERP adalah sebuah sistem diperuntukan bagi perusahaan manufaktur maupun jasa yang berperan mengintegrasikan dan mengotomasikan proses bisnis yang berhubungan dengan aspek operasi, produksi, maupun distribusi di perusahaan bersangkutan [2]. ERP sendiri bermula pada kebutuhan bisnis sebuah perusahaan skala *enterprise* dalam menjalankan semua proses bisnisnya.

Keuntungan penggunaan ERP diantaranya adalah integrasi data keuangan, standarisasi proses operasi, standarisasi data dan informasi, penurunan inventori dan tenaga kerja, peningkatan servis dan kontrol keuangan dan penurunan waktu yang dibutuhkan untuk mendapatkan informasi. Ada pun departemen-departemen atau bagian-bagian yang pada umumnya terintegrasi meliputi:

- a. Pengelolaan keuangan (*financial management*) dalam aplikasi ERP 2016 ini diwakili oleh modul *Account Payable, Account Receivable, Fixed Asset, Cost Management, General Ledger*.
- b. Pengelolaan rantai pemasokan (*supply chain management*) diwakili oleh modul *Purchasing*.
- c. Perencanaan produksi dan manufaktur (*manufacturing resource planning*) diwakili oleh modul *Production dan Inventory*.
- d. Pengelolaan sumber daya manusia (*Human Resource Management*) diwakili oleh modul *Human Resource*.
- e. Pengelolaan relasi dengan pelanggan (*customer relationship management*) diwakili oleh modul *Sales and Distribution*.

## 2.3. Modul Produksi

Proses produksi adalah serangkaian aktivitas dan langkah-langkah untuk menghasilkan suatu barang [2]. Proses produksi disusun oleh *master data* yaitu, *Bill of Material*, dan *Routings*. Sedangkan proses produksi disusun oleh beberapa

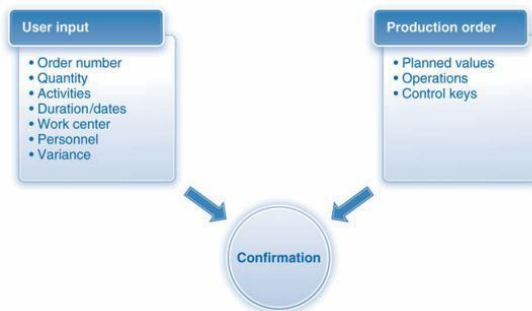
proses utama, yaitu *Regeust for Production*, *Production Order*, *Goods Receive*, *Confirmation*, dan *Goods Transfer*. Keseluruhan proses produksi dijabarkan pada sub bab 2.3.1 hingga 2.3.7.

Fungsi utama kegiatan produksi adalah menghasilkan suatu produk. Untuk dapat menjalankan fungsi tersebut, modul produksi harus berinteraksi dengan modul-modul lainnya, diantaranya adalah modul pergudangan (*Inventory and Warehouse Management*), modul asset (*Asset Management*), dan modul kepegawaian (*Human Resource Management*).

Interaksi modul produksi dengan modul pergudangan digambarkan dengan Gambar 2.1. Permintaan produksi dilakukan oleh modul pergudangan melalui proses *Request Production*. Permintaan tersebut kemudian disetujui oleh modul produksi melalui proses *Authorize Production*. Setelah proses produksi suatu barang dipastikan dimulai maka, modul pergudangan akan mengirimkan barang-barang bahan produksi melalui proses *Issue Raw Material*. Setelah barang bahan produksi telah siap, maka dimulailah proses produksi itu sendiri. Setelah barang telah selesai diproduksi maka barang tersebut akan dikirimkan kembali ke modul pergudangan dengan proses *Receive Finished Goods*. Interaksi modul produksi dengan modul asset dan kepegawaian terjadi pada proses bisnis *confirmation* pada Gambar 2.2. Pada gambar tersebut, proses bisnis *confirmation* melibatkan penggunaan data *personnel* yang didapat dari modul kepegawaian dan data *work center* yang di dalamnya terkandung data mesin. Data mesin pada *work center* didapat melalui modul asset.



**Gambar 2.1 Keterkaitan Antara Modul Produksi dan *Inventory and Warehouse Management***



**Gambar 2.2 Proses Bisnis *Confirmation***

Keterkaitan modul produksi dengan modul-modul lainnya di dalam ERP dirangkum melalui Tabel 2.3 Keterkaitan Antar Modul.

**Tabel 2.3 Keterkaitan Antar Modul**

No.	Proses Bisnis Per Modul				
	Produksi	Pergudangan	Asset	Kepegawaian	Keterangan
1.	<i>Goods Receive</i>	<i>Goods Issue</i>	-	-	Pengiriman barang bahan produksi.
2.	<i>Goods Transfer</i>	<i>Goods Receipt</i>	-	-	Pengiriman barang hasil produksi
3.	<i>Confirmation</i>	-	<i>Asset Depreciation</i>	-	Depresiasi nilai mesin
4.	<i>Confirmation</i>	-	-	<i>Payroll</i>	Penggajian
5.	<i>Production Order</i>	-	<i>Register Asset</i>	<i>Assign Employee</i>	Alokasi sumberdaya produksi.

### **2.3.1. *Bill of Material***

*Bill of Material* adalah daftar lengkap bahan-bahan, baik bahan mentah (*raw material*) maupun bahan setengah jadi (*semifinished goods*), yang merupakan penyusun dari suatu produk (*finished goods*) [2]. *Bill of Material* disusun secara hirarki dengan susunan terbawah berupa *raw material* hingga susunan teratas yang berupa *finished goods*. Data-data penyusun *Bill of Material* adalah ID *finished goods*, deskripsi, dan data setiap material penyusun berupa ID material dan jumlah.

### **2.3.2. *Routings***

*Routings* adalah daftar operasi yang harus dilakukan untuk menghasilkan suatu *semifinished goods* atau *finished goods* [2]. Data-data yang terlibat di dalam *Routings* berupa ID *finished goods*, dan operasi-operasi penyusun.

### **2.3.3. *Request for Production***

*Request for Production* adalah proses permintaan produksi suatu produk [2]. Proses ini terjadi akibat adanya permintaan produksi oleh permintaan pelanggan maupun oleh hasil peramalan penjualan (*sales forecasting*) oleh Modul *Sales and Distribution*. Data-data yang terlibat di dalam *request for production* adalah ID *finished goods*, jumlah, dan tanggal permintaan. *Request for production* merupakan awal dari proses produksi yang membutuhkan persetujuan oleh petugas terkait (*stakeholder*) untuk diteruskan menjadi *Production Order*.

### **2.3.4. *Production Order***

*Production order* merupakan kelanjutan dari proses *request for production* yang telah disetujui (*authorize*) oleh *stakeholder* [2]. Proses ini menandakan mulainya proses produksi yang diawali oleh alokasi bahan baku melalui data *Bill of Material* dan alokasi sumberdaya produksi melalui data *Routings* dan Algoritma Genetika. Data-data yang terlibat di dalam *production order* adalah ID *finished goods*, jumlah, tanggal mulai produksi, dan tanggal selesai produksi.

### **2.3.5. *Goods Receive***

*Goods Receive* adalah proses penerimaan barang mentah produksi dari Modul *Inventory and Warehouse Management* [2]. Barang mentah yang terlibat merupakan material-material penyusun suatu produk yang telah didefinisikan pada *Bill of Material*. Data-data yang terlibat pada proses ini adalah ID *production order*, tanggal penerimaan, dan daftar material yang diterima berupa ID material, dan jumlahnya.

### **2.3.6. *Confirmation***

*Confirmation* adalah proses yang menandakan selesainya penggunaan sumberdaya produksi [2]. Catatan penggunaan sumberdaya manusia kemudian digunakan oleh modul *Human Resource Management* sebagai dasar perhitungan gaji. Sedangkan catatan penggunaan sumberdaya mesin digunakan oleh modul *Asset Management* untuk menghitung depresiasi mesin. Data-data yang terlibat di dalam proses ini adalah ID *employee*, ID mesin, waktu mulai pemakaian, waktu selesai pemakaian, operasi yang terlibat, dan durasi pemakaian sumberdaya.

### **2.3.7. *Goods Transfer***

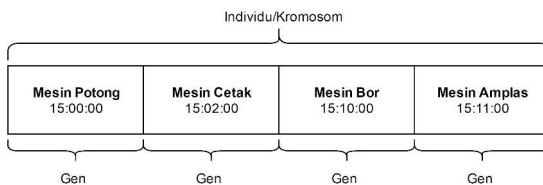
*Goods Transferi* adalah akhir dari proses produksi. Proses ini mendefinisikan barang hasil produksi yang akan dikirimkan kembali ke Modul *Inventory and Warehouse Management* [2]. Data yang terlibat pada proses ini adalah ID material, jumlah, dan tanggal pengiriman.

## **2.4. Algoritma Genetika**

Algoritma Genetika/*Genetic Algorithm* (GA) adalah salah satu algoritma pencarian yang mengikuti konsep seleksi alam di bidang ilmu biologi [3]. Algoritma ini banyak digunakan pada permasalahan optimasi, khususnya permasalahan penjadwalan di dalam modul produksi. GA akan digunakan dalam melakukan pencarian terhadap sumberdaya produksi serta penentuan jadwal produksi.

### 2.4.1. Individu/Kromosom

Di dalam GA, setiap solusi dari permasalahan disebut dengan individu atau kromosom. Kumpulan dari beberapa individu disebut dengan populasi. Sedangkan, individu itu sendiri terdiri atas sejumlah gen. Gen merupakan representasi informasi yang dibawa [3]. Pada Gambar 2.3, individu dinyatakan sebagai solusi penjadwalan produksi. Sedangkan, gen-gen penyusunnya berisi nama mesin dan waktu mulai pekerjaan pada mesin tersebut.



**Gambar 2.3 Susunan Kromosom**

Kromosom/*Chromosome* adalah representasi solusi dari Algoritma Genetika. Kromosom didapatkan berdasarkan urutan operasi produksi dari suatu barang dan sumberdaya pada setiap operasi. Kromosom disusun dari gen-gen yang membawa informasi. Pada kasus penjadwalan, gen membawa informasi mengenai pekerja, mesin, dan waktu operasi seperti yang dijelaskan pada Tabel 2.4 Sumberdaya per Operasi.

**Tabel 2.4 Sumberdaya per Operasi**

Mempersiapkan Material	Perakitan Rangka	Pengecetan	Pengemasan
ID Sumberdaya : 1 Pegawai : Hadi Mesin : - Waktu : 1 menit	ID Sumberdaya : 3 Pegawai : Budi Mesin : Welding #1 Waktu : 3 menit	ID Sumberdaya : 5 Pegawai : Steve Mesin : Spray #1 Waktu : 5 menit	ID Sumberdaya : 7 Pegawai : Ainun Mesin : - Waktu : 2 menit



<b>Mempersiapkan Material</b>	<b>Perakitan Rangka</b>	<b>Pengecetan</b>	<b>Pengemasan</b>
ID Sumberdaya : 2 Pegawai : Sunarto Mesin : - Waktu : 1 menit	ID Sumberdaya : 4 Pegawai : Maman Mesin : Welding #2 Waktu : 4 menit	ID Sumberdaya : 6 Pegawai : Koko Mesin : Spray #2 Waktu : 3 menit	ID Sumberdaya : 8 Pegawai : Agus Mesin : - Waktu : 1 menit

Pada Tabel 2.5 Contoh Alokasi Sumberdaya per Operasi dijabarkan contoh alokasi sumberdaya manusia, mesin, dan waktu operasi untuk masing-masing operasi.

**Tabel 2.5 Contoh Alokasi Sumberdaya per Operasi**

	<b>Mempersiapkan Material</b>	<b>Perakitan Rangka</b>	<b>Pengecetan</b>	<b>Pengemasan</b>
<i>Job 1</i>	Sumberdaya 1	Sumberdaya 3	Sumberdaya 6	Sumberdaya 7
<i>Job 2</i>	Sumberdaya 2	Sumberdaya 4	Sumberdaya 5	Sumberdaya 7
<i>Job 3</i>	Sumberdaya 2	Sumberdaya 3	Sumberdaya 5	Sumberdaya 8

Urutan operasi dengan alokasi sumberdaya untuk produksi sebuah barang disebut dengan *job*. Jika dalam satu kali proses produksi akan diporduksi 10 buah barang, maka akan terdapat 10 buah *job*. Gambar 2.7 merupakan contoh bentuk kromosom dari *job-job* pada Tabel 2.4 *Gen (1,1)* merupakan representasi dari *job 1* dan operasi ke 1.

(1,1)	(2,1)	(1,2)	(3,1)	(3,2)	(1,3)	(2,2)	(2,3)	(3,3)	(1,4)	(2,4)	(3,4)
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

**Gambar 2.4 Contoh Kromosom**

Penyusunan kromosom pada Gambar 2.4 menggunakan prosedur *non-delay permutation encoding* [5]. Berikut tahap-tahap *non-delay permutation encoding* :

1. Buat himpunan operasi yang mengawali pekerjaan :  
 $A = \{o_{il} \mid 1 \leq i \leq n\}$
2.
  - i. Pilih  $o^l$ , operasi dengan waktu selesai tercepat,  
 $t^l + p^l \leq t_{ik} + p_{ik}$  untuk semua  $o_{ik} \in A$ . Jika terdapat lebih dari satu operasi, maka pilih operasi yang terletak paling kiri.
  - ii. Jika  $M^l$  adalah sumberdaya yang dipakai oleh  $o^l$ , maka buatlah himpunan  $B$  yang berisi semua operasi dari  $A$  yang diproses di  $M^l$ .
  - iii. Pilih  $o^{ll}$ , operasi dengan waktu mulai paling awal di  $B$ . Jika terdapat lebih dari satu operasi, maka pilih operasi yang terletak paling kiri.
  - iv. Pilih operasi di  $B$  yang terletak paling kiri dan hapus dari  $A$ . Operasi yang dipilih tersebut adalah  $o_{ik}^*$ .
3. Masukkan operasi  $o_{ik}^*$  pada jadwal serta hitung waktu mulai operasi tersebut :  $t_{ik}^* = \max(t_{i,k-1}^* + p_{i,k-1}^*, t_{hl} + p_{hl})$ , dimana  $t_{ik}^*$  adalah waktu mulai operasi  $o_{ik}^*$ ,  $t_{i,k-1}^* + p_{i,k-1}^*$  adalah waktu selesai operasi ke  $k-1$  yang merupakan operasi sebelumnya dari *job* ke- $i$  dan  $t_{hl} + p_{hl}$  adalah waktu selesai operasi ke- $l$  dan *job* ke- $h$  yang mendahului  $o_{ik}^*$  pada sumberdaya yang sama.
4. Jika terdapat suksesor dari  $o_{ik}^*$ , yaitu  $o_{i,k+1}^*$  maka tambahkan ke  $A$ .
5. Ulangi langkah 2 sampai  $A$  telah kosong.

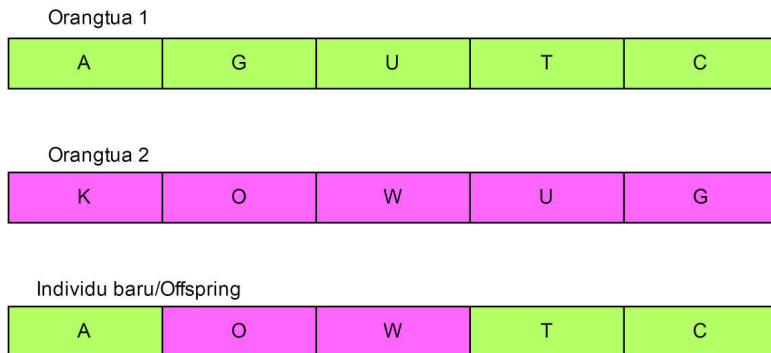
dimana,

$o_{ik}$  = operasi ke- $k$  dari *job* ke- $i$ ,  
 $o_{il}$  = operasi pertama dari *job* ke- $i$ ,  
 $t_{ik}$  = waktu mulai operasi ke- $k$  dari *job* ke- $i$ ,  
 $p_{ik}$  = waktu proses operasi ke- $k$  dari *job* ke- $i$

Langkah awal dari Algoritma Genetika adalah pembentukan calon-calon solusi secara acak sejumlah  $n$  kromosom. Jumlah kromosom pada sebuah populasi biasanya sekitar 20-30 kromosom.

#### 2.4.2. Persilangan/*Crossover*

Persilangan/*Crossover* adalah proses pembuatan individu baru (*offspring*) dari 2 individu orang tua. Sifat/gen yang dimiliki orang masing-masing orang tua akan diturunkan pada keturunannya [3]. Pewarisan sifat dicontohkan seperti pada Gambar 2.5.



**Gambar 2.5 Proses Persilangan**

Pindah Silang/*Crossover* adalah proses pembuatan kromosom baru dari 2 induk yang dipilih melalui metode *Roulette Wheel Selection* di dalam populasi. Kromosom baru hasil *crossover* akan dijadikan individu baru di populasi/generasi selanjutnya. Pindah silang menggunakan metode *Precedence Preservative Crossover* (PPX) [5]. Pada Gambar 2.5 Proses Persilangan dijelaskan metode PPX yang akan digunakan pada permasalahan penjadwalan. Sejumlah  $n$  *gen* bilangan bulat acak antara 1 dan 2 dibangkitkan. Untuk setiap angka 1 dari bilangan

bulat acak yang dihasilkan lakukan pewarisan *gen* dari kromosom orangtua 1 kepada kromosom individu baru, sebaliknya untuk setiap angka 2 dari bilangan bulat acak yang dihasilkan lakukan pewarisan *gen* dari kromosom orangtua 2 kepada kromosom individu baru.

### 2.4.3. *Fitness Function*

*Fitness function* adalah nilai yang dimiliki oleh setiap individu untuk tetap bertahan terhadap seleksi yang terjadi [3]. Umumnya, semakin besar *fitness function* yang dimiliki oleh sebuah individu, maka semakin besar pula peluang individu tersebut untuk tetap bertahan sebagai solusi. Karena GA adalah algoritma optimasi, sedangkan permasalahan penjadwalan adalah proses minimasi, maka *fitness function* akan membagi angka 1 dengan *average flow time* dari kromosom yaitu  $f = 1 / t$ , dimana,  $f$  = *fitness value* dan  $t$  = *average flow time* [3].

### 2.4.4. Mutasi

Mutasi adalah proses perubahan struktur kromosom pada skala *gen* [3]. Metode mutasi yang digunakan adalah metode *remove and insert* [5]. 2 *gen* di dalam sebuah kromosom akan dipilih secara acak dan dilakukan pertukaran posisi *gen* antar 2 *gen* tersebut.

## 2.5. *Multitenant Data Architecture* atau *Multitenancy*

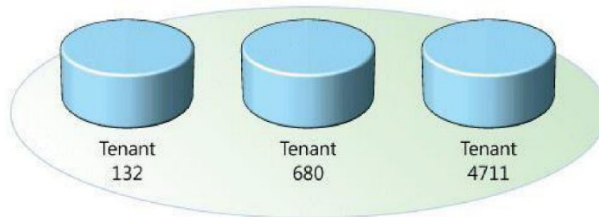
*Multitenancy* mengacu pada arsitektur perangkat lunak di mana salah satu contoh dari perangkat lunak yang berjalan pada server dan menyajikan beberapa *tenant* [1]. *Tenant* adalah sekelompok pengguna yang berbagi akses umum dengan hak akses khusus untuk contoh perangkat lunak. Dengan arsitektur *multitenant*, aplikasi perangkat lunak ini dirancang untuk memberikan setiap *tenant* sebuah bagian untuk berdedikasi contoh termasuk data, konfigurasi, manajemen pengguna, fungsi individu *tenant* dan sifat non-fungsional. *Multitenancy* bertentangan dengan arsitektur *multi-instance* misalnya, di mana

contoh perangkat lunak terpisah beroperasi atas nama penyewa yang berbeda.

Arsitektur data *Multitenancy* digunakan karena dinilai memadai dan cukup aman dalam mengatasi masalah kurang kepercayaan *tenant* untuk menyerahkan kontrol data bisnis *tenant* kepada pihak ketiga. Terdapat 3 jenis *Multitenancy*, antara lain:

### 2.5.1. Separated Database

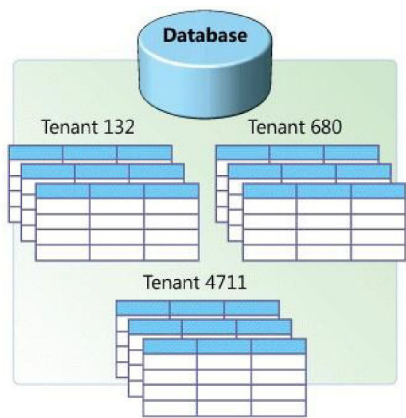
Pada multitenant separated database, setiap tenant mendapatkan masing-masing database, sehingga mudah untuk mengatur kembali model data aplikasi yang digunakan [1]. Akan tetapi dalam penggunaan arsitektur ini memerlukan biaya yang tinggi untuk menjaga peralatan yang digunakan untuk server dan juga untuk back up data dari masing-masing *tenant* seperti pada Gambar 2.6.



**Gambar 2.6 Separated Schema, Separated Database**

### 2.5.2. Shared Database, Separated Schema

Pada arsitektur multitenant shared database, separate schema, saat tenant menyewa ke servis, subsistem yang ada akan membuat tabel untuk tenant sesuai dengan skema yang dimiliki [1]. Arsitektur ini mudah untuk digunakan (tabel yang dibentuk pada awalnya merupakan tabel standar, dan selanjutnya dapat diubah sesuai keinginan tenant), akan tetapi dalam apabila terjadi kegagalan, maka perlu dilakukan perbaikan untuk semua tabel yang ada di dalam *database* seperti pada Gambar 2.7.



Gambar 2.7 *Shared Database, Separated Schema*

2.5.3. Shared Database, Shared Schema

Pada arsitektur multitenant *shared database, shared schema*, menggunakan *database* dan tabel yang sama untuk banyak tenant, untuk membedakan masing-masing tenant digunakan *id tenant* untuk setiap tabel yang ada [1]. Penggunaan model ini tidak memerlukan biaya yang tinggi, akan tetapi apabila terjadi kegagalan, akan sama seperti dengan model multitenant *shared database, separate schema*. Model ini menggunakan mekanisme seperti pada Gambar 2.8.

	TenantID	CustName	Address
4	TenantID	ProductID	ProductNam
1	4	TenantID	Shipment
6	1	4711	324965
			2006-02-21
4	6	132	115468
			2006-04-08
	4	680	654109
			2006-03-27
		4711	324956
			2006-02-23

Gambar 2.8 *Shared Schema, Shared Database*

## 2.6. *Distributed Database*

Basis data terdistribusi (BDT) adalah suatu basis data yang memiliki kontrol terpusat pada *server* manajemen dengan distribusi penyimpanan data yang tersebar di beberapa *server* data [4]. BDT diterapkan dalam rangka untuk menjaga prinsip *high availability* pada aplikasi ERP. DBT memungkinkan aplikasi tetap berjalan jika terdapat sebuah *server* basis data mengalami *system failure*. Terdapat 2 mekanisme utama dalam BDT, yaitu replikasi dan fragmentasi.

### 2.6.1. Replikasi

Replikasi adalah mekanisme penyalinan seluruh atau sebagian *table* basis data ke beberapa *server data* [4]. Setiap transaksi (penambahan, penghapusan, atau perubahan data) akan dieksekusi pada semua *server* data yang menyusun sistem BDT. Hal ini menyebabkan mekanisme replikasi membebani performa sistem, namun memiliki tingkat kompleksitas yang paling sederhana.

### 2.6.2. Fragmentasi

Fragmentasi adalah mekanisme penyalinan sebagian data atau struktur dari setiap tabel basis data [4]. Transaksi basis data yang terjadi harus diolah terlebih dahulu oleh *server* manajemen untuk menentukan letak *server* data dari tabel yang akan terpengaruh. Hal ini menyebabkan mekanisme fragmentasi memiliki tingkat kompleksitas yang tinggi, namun memiliki performa yang lebih baik dibandingkan dengan mekanisme replikasi

## 2.7. *MySQL Cluster*

*MySQL Cluster* adalah salah satu RDBMS (*Relational Database Management System*) yang memungkinkan klusterisasi basis data [7]. Hal ini dilakukan dengan memecah basis data ke beberapa *server* fisik berbentuk *node-node mysql* yang saling terintegrasi satu sama lain. Hal ini biasanya diperlukan untuk

pembuatan basis data terdistribusi dimana memiliki kelebihan replikasi dan *high-availability* pada sistem database.

## 2.8. PHP (Hypertext Preprocessor)

PHP adalah singkatan dari Hypertext Preprocessor yakni salah satu bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML [6]. PHP diciptakan oleh Rasmus Lerdorf pertama kali tahun 1994. Pada awalnya PHP adalah singkatan dari *Personal Home Page Tools*. Selanjutnya diganti FI (*Forms Interpreter*). Sejak versi 3.0, nama bahasa ini diubah menjadi PHP (*Hypertext Preprocessor*) dengan singkatan PHP. PHP versi terbaru adalah versi ke-7.

## 2.9. Framework Yii 2

Sebuah *framework* untuk bahasa pemrograman PHP dimana framework ini telah didukung oleh aspek *modularity* dan *eloquent database* [8]. Arsitektur perangkat lunak ini menggunakan *Model-View-Controller*, namun dengan aspek *modularity*, maka dapat dibuat modul dimana setiap modul memiliki MVC tersendiri. Sedangkan *eloquent database* adalah basisdata berupa objek di dalam sebuah bahasa pemrograman PHP.



## **BAB III**

### **ANALISIS DAN PERANCANGAN SISTEM**

Bab ini membahas tahap analisis dan perancangan sistem yang akan dibangun. Analisis membahas semua persiapan yang akan menjadi pokok pikiran pembuatan aplikasi ini. Mulai dari masalah yang melatarbelakangi, hingga analisis gambaran awal sistem yang akan dibuat. Perancangan sistem membahas hal-hal yang berkaitan dengan pondasi atau dasar pembuatan aplikasi, yang meliputi perancangan basis data, tampilan antar muka halaman aplikasi, hingga perancangan alur proses yang akan diimplementasikan di dalam aplikasi.

#### **3.1. Analisis**

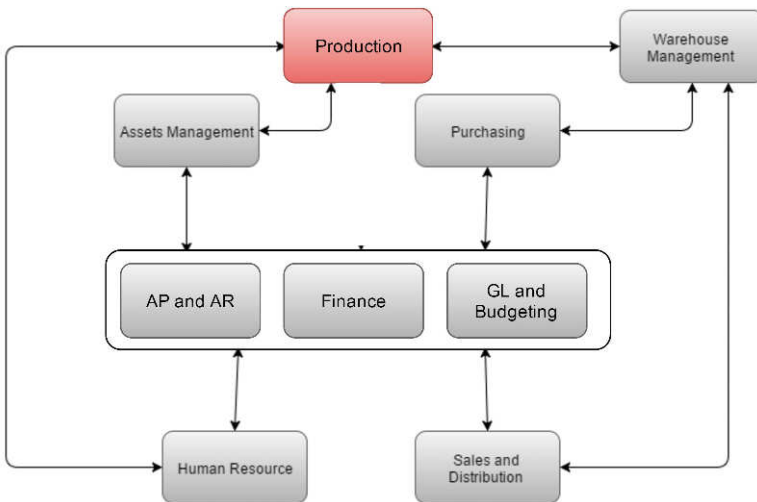
Tahap analisis dibagi menjadi beberapa bagian antara lain: domain permasalahan, deskripsi umum sistem, dan kasus penggunaan sistem. Berikut pembahasan bagian-bagian tahap analisis.

##### **3.1.1. Analisis Proses Bisnis**

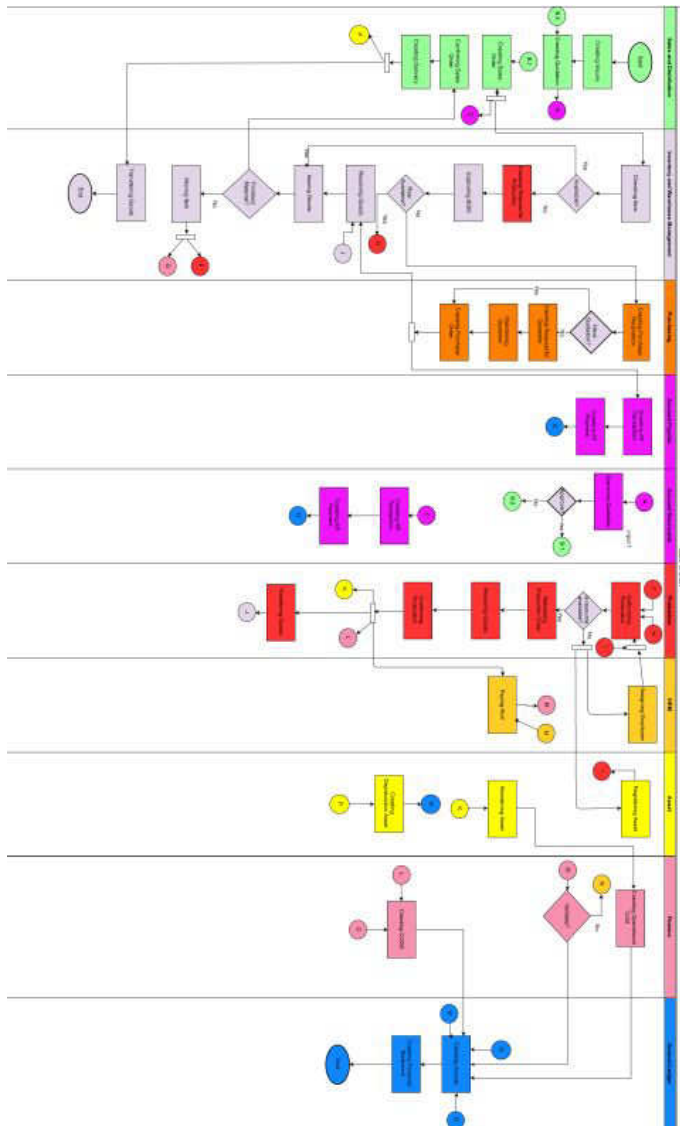
Berdasarkan subbab 2.2 mengenai sistem ERP, maka didapatkan gambaran umum proses bisnis ERP. Gambaran umum tersebut dapat dilihat pada Gambar 3.1 dengan Modul Produksi berwarna merah. Modul-modul keuangan (AP/AR, *Finance*, dan GL /*Budgeting*) merupakan sentral dari sistem ERP. Sedangkan, modul produksi berinteraksi dengan modul Pergudangan (*Inventory and Warehouse Management*), Modul Aset (*Assets Management*), dan Modul Kepegawaian (*Human Resource*).

Gambaran umum proses bisnis ERP yang telah terbentuk kemudian dipecah menjadi proses bisnis - proses bisnis yang lebih rinci. Hal tersebut dapat dilihat dari Gambar 3.2 dan Gambar 3.3. Pemecahan gambaran umum proses bisnis ERP didasarkan pada subbab 2.3. Hasil pemecahan Modul Produksi adalah proses bisnis *Request for Production*, *Production Order (Authorizing Production)* dan *Releasing Production*, *Receiving Goods*,

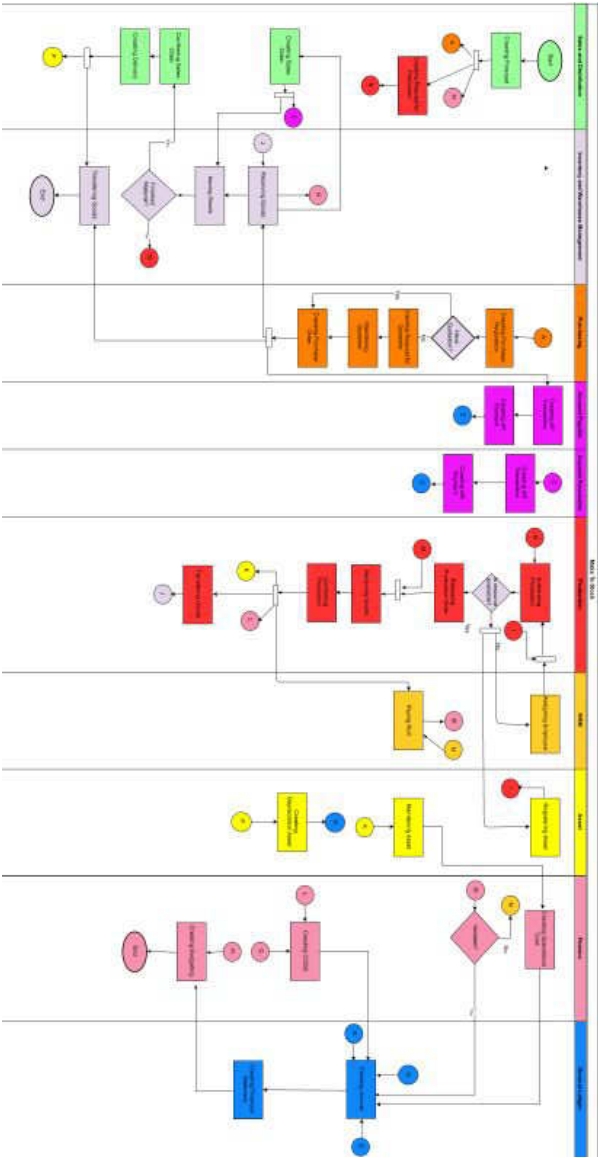
*Confirming Production*, dan *Transferring Goods*. Berdasarkan tabel 2.2 pada subbab 2.2 terdapat keterkaitan antara beberapa proses bisnis modul produksi dengan beberapa proses bisnis modul pergudangan, aset, dan kepegawaian. Keterkaitan proses bisnis antar modul tersebut diantaranya, *Goods Receive* modul production dengan *Goods Issue* modul pergudangan, *Goods Transfer* dengan *Goods Receipt* modul pergudangan, *Confirmation* dengan *Asset Depreciation* modul aset, *Confirmation* dengan *Payroll* modul kepegawaian, *Production Order* dengan *Register Asset* modul aset, dan *Production Order* dengan *Assign Employee* modul kepegawaian. Keterkaitan ini tampak pada proses bisnis level 1 yaitu pada Gambar 3.2 dan Gambar 3.3. Terdapat dua jenis skema bisnis utama yang digunakan, yakni *Make-To-Order* (MTO) yang ditunjukkan oleh gambar 3.2 dan *Make-To-Stock* (MTS) yang ditunjukkan oleh gambar 3.3. MTO adalah sebuah skema dimana produksi didasarkan pada permintaan. MTS adalah sebuah skema dimana produksi didasarkan oleh *sales-forecasting*.



**Gambar 3.1 Gambaran Umum Proses Bisnis ERP**



**Gambar 3.2 Skema *Make to Order***

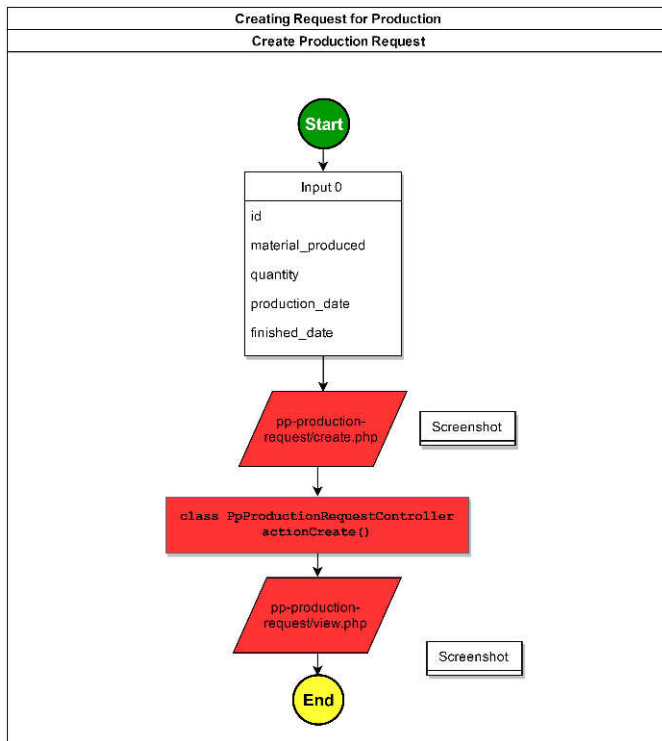


Gambar 3.3 Skema *Make to Stock*

Berdasarkan skema MTO dan MTS, modul produksi memiliki beberapa proses bisnis utama, yaitu *Creating Request for Production*, *Authorizing Production*, *Releasing Production Order*, *Receiving Goods*, *Confirming Production*, dan *Transferring Goods*. Maka, analisa dan perancangan setiap proses bisnis adalah sebagai berikut :

### 3.1.1.1. *Creating Request for Production*

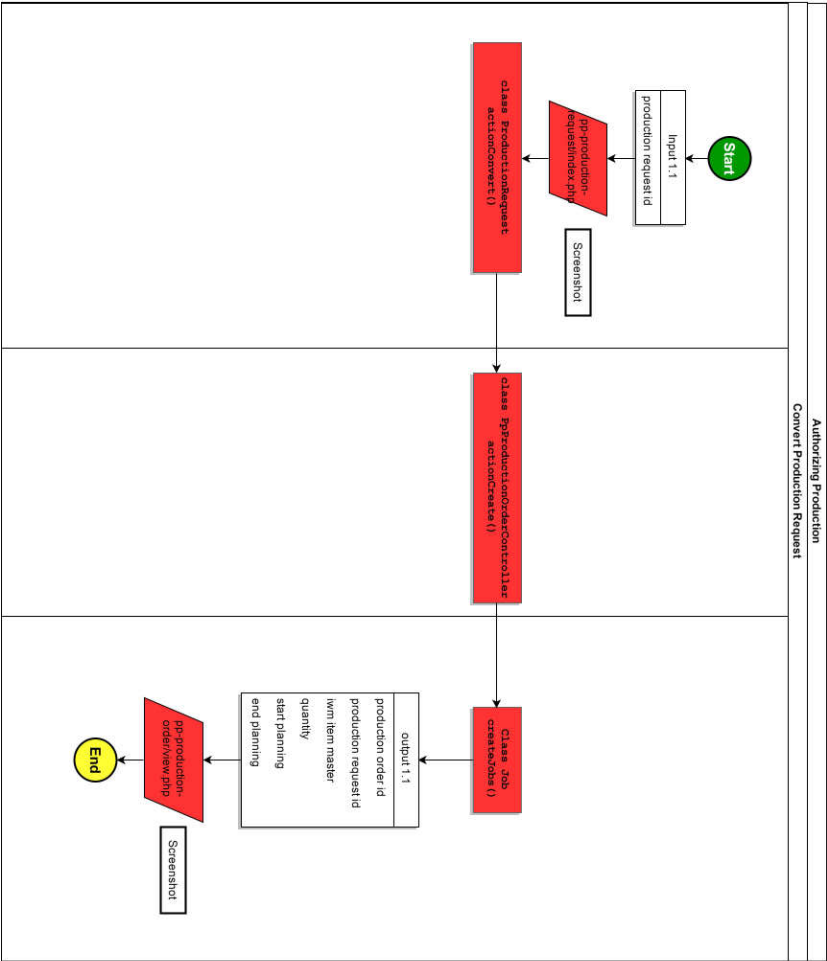
*Creating Request for Production* pada Gambar 3.4 mewakili *Request for Production* pada subbab 2.3.3.



**Gambar 3.4** Penjabaran *Creating Request for Production*

3.1.1.2. *Authorizing Production*

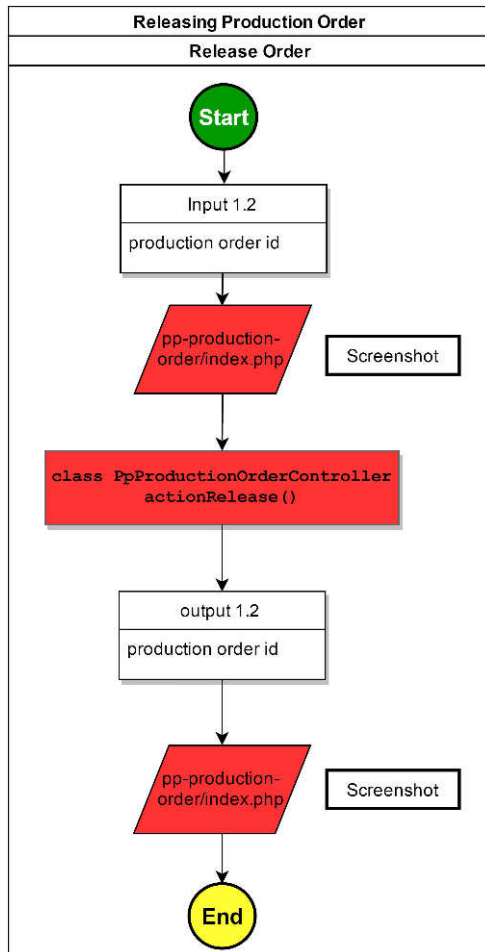
*Authorizing Production* pada Gambar 3.5 adalah proses pembuatan *Production Order* pada subbab 2.3.4.



Gambar 3.5 Penjabaran *Authorizing Production*

### 3.1.1.3. *Releasing Production Order*

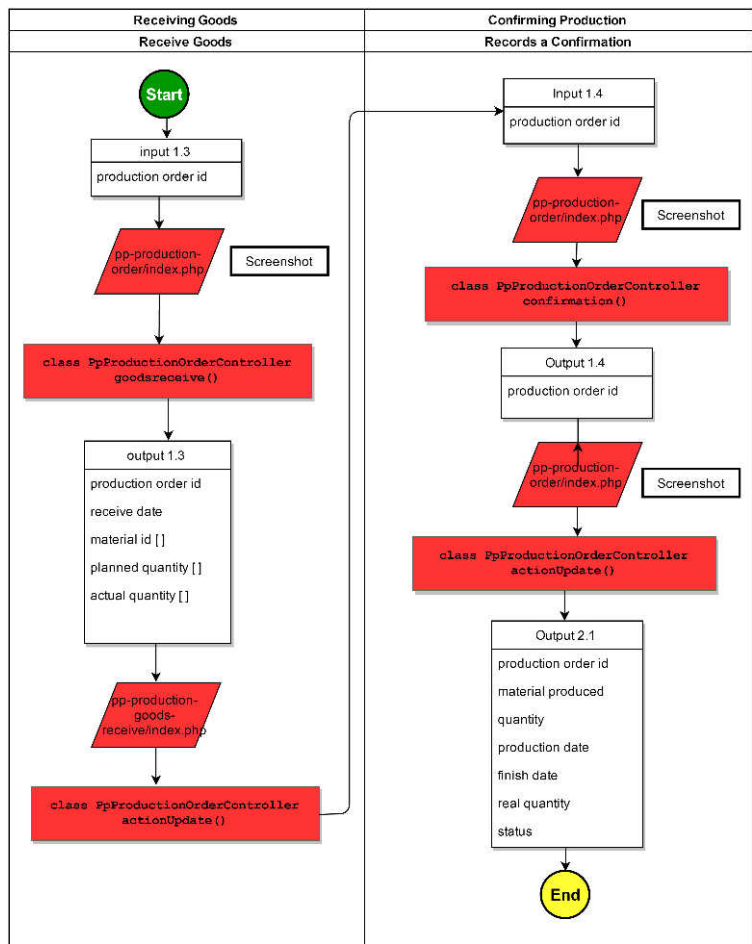
*Releasing Production Order* pada Gambar 3.6 adalah proses perubahan status *Production Order* pada subbab 2.3.4 sebagai tanda dimulainya produksi.



**Gambar 3.6** Penjabaran *Releasing Production Order*

3.1.1.4. *Receiving Goods dan Confirming Production*

*Receiving Goods dan Confirming Production* pada Gambar 3.7 Mewakili proses pada subbab 2.3.5 dan subbab 2.3.6.

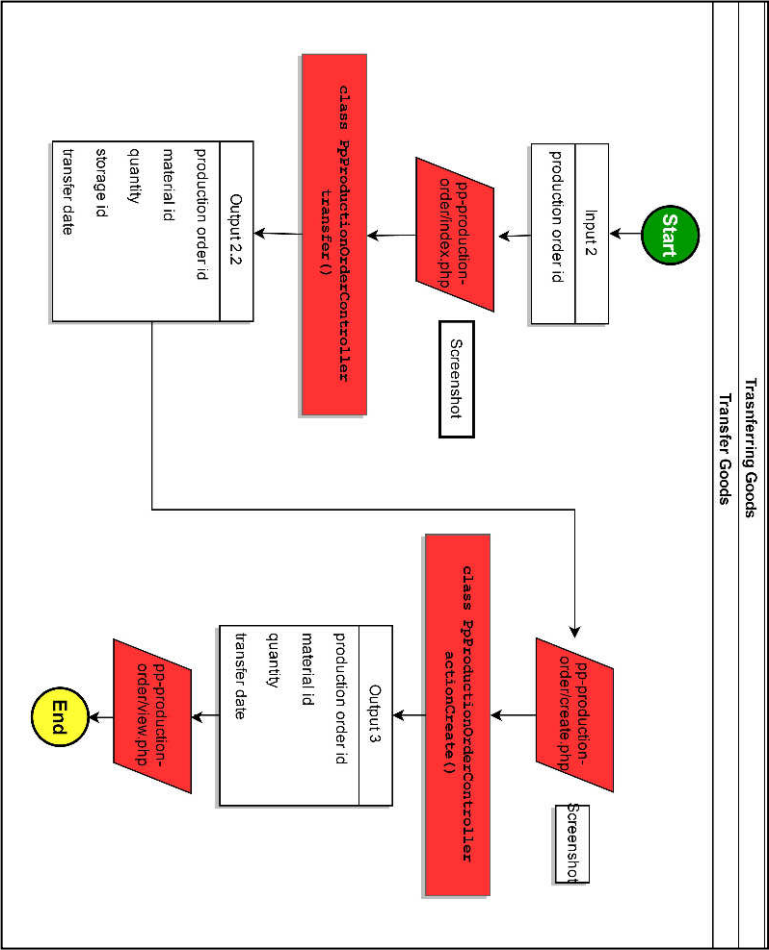


Gambar 3.7 Penjabaran *Receiving Goods*



3.1.1.5. *Transferring Goods*

*Transferring Goods* pada Gambar mewakili *Goods Transfer* pada subbab 2.3.7.



Gambar 3.8 Penjabaran *Transferring Goods*

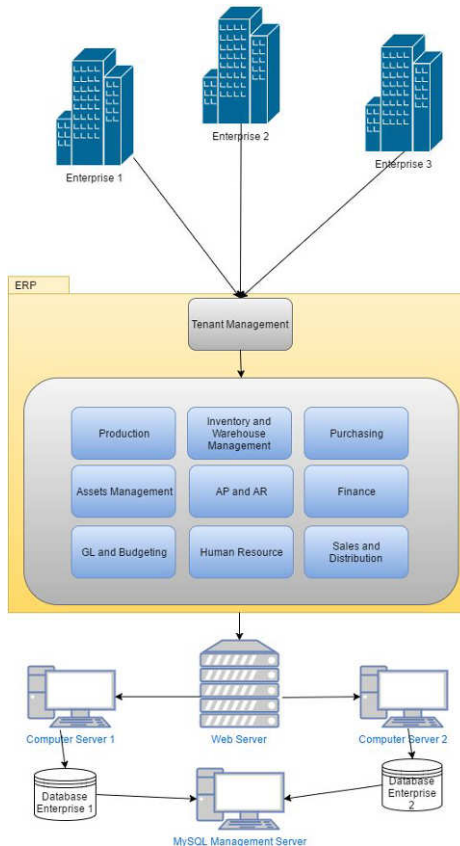
### 3.1.2. Analisis Data

Konsep autentikasi dan otorisasi data yang digunakan adalah *multi-tenancy separated database* seperti yang dijelaskan pada sub bab 2.5.1. Pada konsep ini, data milik setiap *tenant* akan disimpan di basis data yang terpisah. Sehingga tidak ada 2 atau lebih *tenant* menggunakan *table* dan basis data yang sama. Penambahan *tenant* baru akan menyebabkan penambahan sebuah basis data baru. Penggunaan konsep *separated database* dimaksudkan agar data milik setiap *tenant* dapat terjamin kerahasiaannya. Keuntungan lainnya adalah kerumitan proses *maintenance* basis data dapat dikurangi.

Sedangkan konsep basis data terdistribusi yang diterapkan adalah replikasi seperti yang dijelaskan pada sub bab 2.6.1. Replikasi memungkinkan penyalinan setiap tabel basis data ke *node-node* penyusun basis data terdistribusi. Kegagalan sebuah *node* tidak akan menyebabkan basis data berhenti bekerja. Sebaliknya, hal tersebut akan memicu mekanisme sinkronisasi jika *node* yang mati kembali hidup. Pada kegagalan suatu *node*, penyimpanan data dilakukan pada *node* lainnya yang masih hidup. Pengambil keputusan mengenai pada *node* mana suatu data akan disimpan dilakukan oleh sebuah *management node*. *Management node* memiliki *log* setiap aktivitas basis data yang terjadi. Berberkalkan *log* tersebut, *management node* akan melakukan operasi-operasi *query* pada *node* yang kembali hidup. Sehingga terciptalah suatu sistem basis data terdistribusi yang mampu menangani kegagalan sebuah *node* data.

### 3.2. Deskripsi Umum Sistem

Pada Tugas Akhir ini dibangun aplikasi Modul Produksi yang terintegrasi ke dalam sebuah sistem *Enterprise Resource Planning* (ERP). Tujuan dari aplikasi Produksi adalah untuk melakukan manajemen terhadap proses produksi. Sistem yang dibangun berorientasi *multi-tenancy* dengan basis data terdisitribusi seperti pada Gambar 3.19.



**Gambar 3.9 Gambaran Umum Sistem**

### 3.3. Aktor

Pada sistem yang akan dibangun, aktor yang akan menjadi pengguna sistem adalah staff produksi dan manajer produksi. Staff dan manajer produksi memiliki hak akses dalam pengelolaan data-data transaksi, sedangkan hak akses pengelolaan data-data master hanya dimiliki oleh manajer produksi.

### 3.4. Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan dalam sistem ini mencakup kebutuhan fungsional. Kebutuhan fungsional berisikan proses-proses yang dibutuhkan dalam sistem dan harus dijalankan. Kebutuhan fungsional sistem dideskripsikan dalam Tabel 3.1

**Tabel 3.1 Kebutuhan Fungsional Sistem**

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
F-001	Mengelola <i>Routings</i>	Pengguna dapat mengelola data routings yang meliputi proses: <i>create, update, delete</i> .
F-002	Mengelola Operasi	Pengguna dapat mengelola data master operasi yang meliputi proses: <i>create, update, delete</i> .
F-003	Mengelola <i>Bill of Material</i>	Pengguna dapat mengelola data master <i>Bill of Material</i> yang meliputi proses: <i>create, update, delete</i> .
F-004	Membuat <i>Request for Production</i>	Pengguna dapat menambahkan data transaksi <i>request for production</i> .
F-005	Melakukan <i>Confirmation</i>	Pengguna dapat melakukan <i>confirmation</i> data <i>production order</i> .
F-006	Membuat <i>Goods Receive</i>	Pengguna dapat menambahkan data transaksi <i>goods receive</i> .
F-007	Membuat <i>Goods Transfer</i>	Pengguna dapat menambah data transaksi

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
		<i>goods transfer.</i>
F-008	<i>Generating Scheduling</i>	Pengguna dapat membuat <i>production order</i> dari <i>request for production</i> yang ada.

### 3.5. Physical Data Model

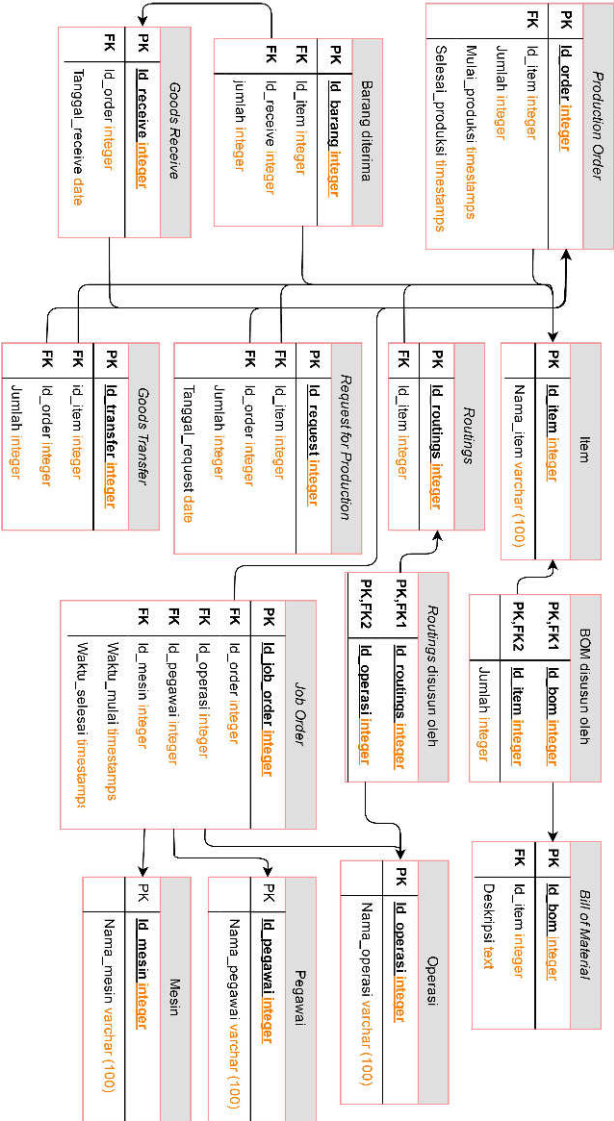
Perancangan *Physical Data Model* (PDM) dari sistem dijelaskan pada Gambar 3.9 dan dijelaskan pada Tabel 3.2. Rincian setiap *table* pada PDM dilampirkan pada Lampiran A.1 hingga A.14.

**Tabel 3.2 Keterangan Tabel pada PDM**

No.	Nama Tabel	Atribut	Deskripsi
1.	Item	Id_item : integer	Tabel untuk menyimpan data barang.
		Nama_item : varchar (100)	
2.	<i>Bill of Material</i>	Id_bom : integer	Tabel untuk menyimpan data <i>Bill of Material</i> .
		Id_item : integer	
		Deskripsi : text	
3.	Pegawai	Id_pegawai : integer	Tabel untuk menyimpan data pegawai.
		Nama_pegawai : varchar (100)	
4.	<i>Routings</i>	Id_routings : integer	Tabel untuk menyimpan data <i>Routings</i> .
		Id_item : integer	
5.	Operasi	Id_operasi : integer	Tabel untuk menyimpan data Operasi.
		Nama_operasi : varchar (100)	

No.	Nama Tabel	Atribut	Deskripsi
6.	Mesin	Id_mesin : int	Tabel untuk menyimpan data mesin.
		Nama_mesin : varchar (100)	
7.	<i>Request for Production</i>	Id_request : integer	Tabel untuk menyimpan data <i>Request for Production</i> .
		Id_order : integer	
		Id_item : integer	
		Jumlah : integer	
		Tanggal_request : date	
8.	<i>Production Order</i>	Id_order : integer	Tabel untuk menyimpan data <i>Production Order</i> .
		Id_item : integer	
		Jumlah : integer	
		Mulai produksi	
		Selesai produksi	
9.	<i>Job Order</i>	Id_job_order : integer	Tabel untuk menyimpan data <i>Scheduling</i> .
		Id_pegawai : integer	
		Id_order : integer	
		Id_mesin : integer	
		Id_operasi : integer	
		Waktu_mulai : timestamps	
		Waktu_selesai : timestamps	
10.	<i>Goods Receive</i>	Id_receive : integer	Tabel untuk menyimpan data <i>Goods Receive</i> .
		Id_order : integer	
		Tanggal_receive : date	
11.	<i>Goods Transfer</i>	Id_transfer :	Tabel untuk

No.	Nama Tabel	Atribut	Deskripsi
		integer	menyimpan data <i>Goods Transfer</i> .
		Id_order : integer	
		Id_item : integer	
		Tanggal_transfer : date	
		Jumlah : integer	
12.	Bom disusun oleh	Id_item : integer	Tabel untuk menyimpan material penyusun BOM.
		Id_bom : integer	
		Jumlah : integer	
13.	<i>Routings</i> disusun oleh	Id_routings : integer	Tabel untuk menyimpan data penyusun <i>Routings</i> .
		Id_operasi : integer	
14.	Barang diterima	Id_barang : integer	Tabel untuk menyimpan data material yang diterima dari pergudangan.
		Id_item : integer	
		Id_receive : integer	
		Jumlah : integer	

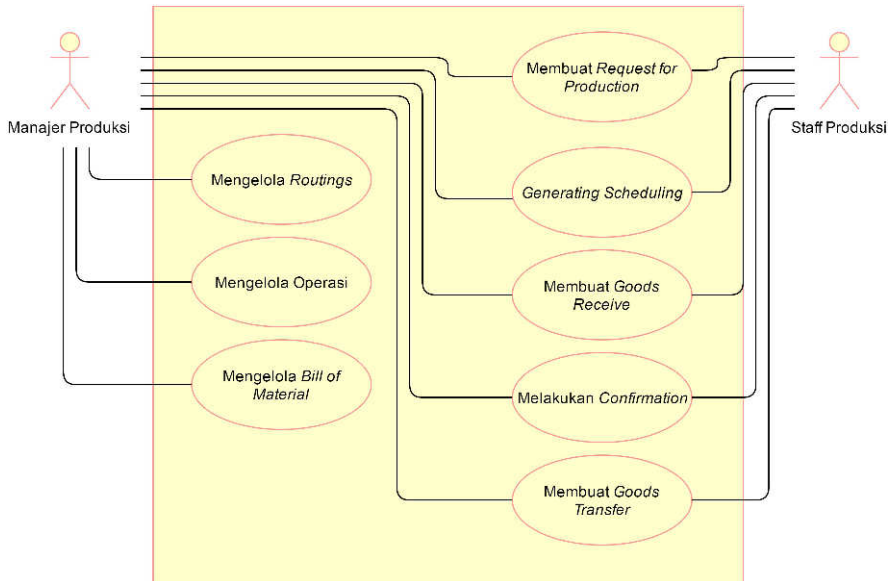


Gambar 3.10 Physical Data Model (PDM)



### 3.6. Kasus Penggunaan

Kasus penggunaan yang dibutuhkan pada sistem sesuai dengan analisa yang telah dilakukan. Diagram kasus penggunaan dapat dilihat pada Gambar 3.10 dan kode kasus penggunaan ada pada tabel 3.3.



**Gambar 3.11 Diagram Kasus Penggunaan**

**Tabel 3.3 Tabel Kode Kasus Penggunaan**

Kode Kasus Penggunaan	Kasus Penggunaan
UC-001	Mengelola <i>Routings</i>
UC-002	Mengelola Operasi
UC-003	Mengelola <i>Bill of Material</i>

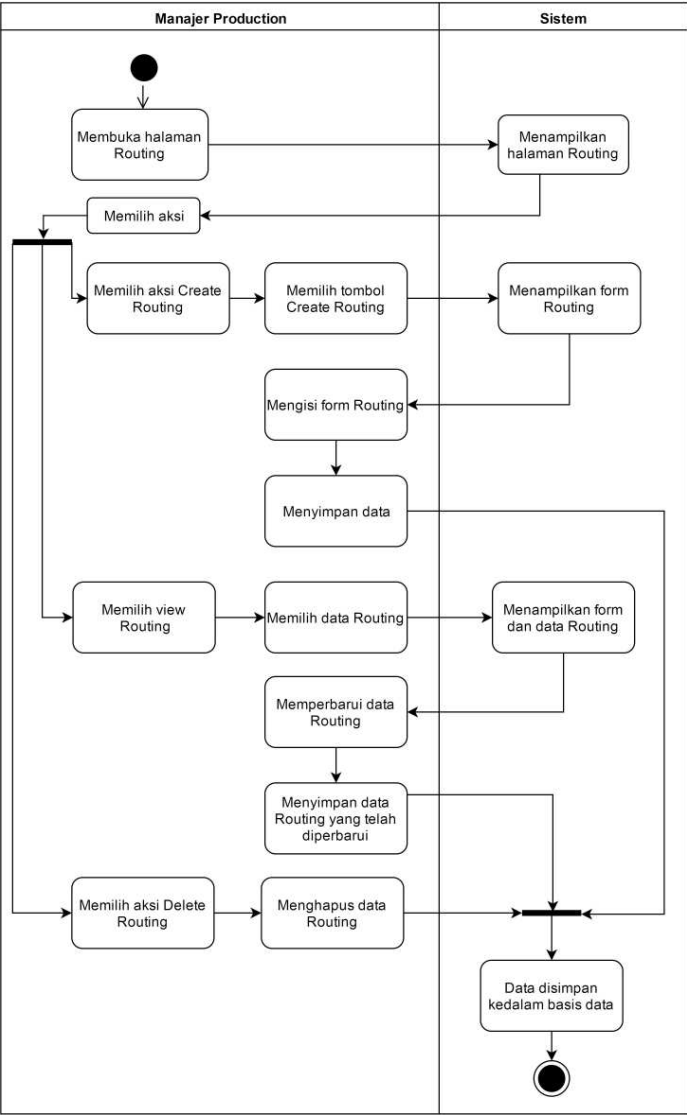
UC-004	Membuat <i>Request for Production</i>
UC-005	Melakukan <i>Confirmation</i>
UC-006	Membuat <i>Goods Receive</i>
UC-007	Membuat <i>Goods Transfer</i>
UC-008	<i>Generating Scheduling</i>

### 3.6.1. Analisis Kasus Penggunaan Mengelola *Routings*

Fungsionalitas mengelola *Routings* digunakan sebagai sarana pengelolaan data master *Routings*. Pengelolaan data master routings hanya bisa dilakukan oleh manajer produksi. Alur aktivitas kasus penggunaan mengelola *Routings* ditunjukkan pada Gambar 3.11.

<b>Nama</b>	Mengelola Routings
<b>Kode</b>	F-001
<b>Deskripsi</b>	Mengelola data master <i>Routings</i>
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Manajer produksi membuka halaman pengelolaan routings.
<b>Aktor</b>	Manajer dan staff produksi
<b>Aliran:</b>	
- <b>Kejadian Normal</b>	<ol style="list-style-type: none"> <li>1. Manajer produksi membuka halaman pengelolaan <i>Routings</i>.</li> <li>2. Sistem menampilkan halaman daftar <i>Routings</i>.</li> <li>3. Manajer produksi memilih pilihan tambah, ubah, atau hapus pada data <i>Routings</i>.</li> <li>4. Manajer produksi melakukan penambahan, perubahan, atau penghapusan data.</li> </ol>

<b>- Kejadian Alternatif</b>	Tidak ada
<b>Kondisi Akhir</b>	Sistem menampilkan daftar data <i>Routings</i> .
<b>Kebutuhan Khusus</b>	Tidak ada

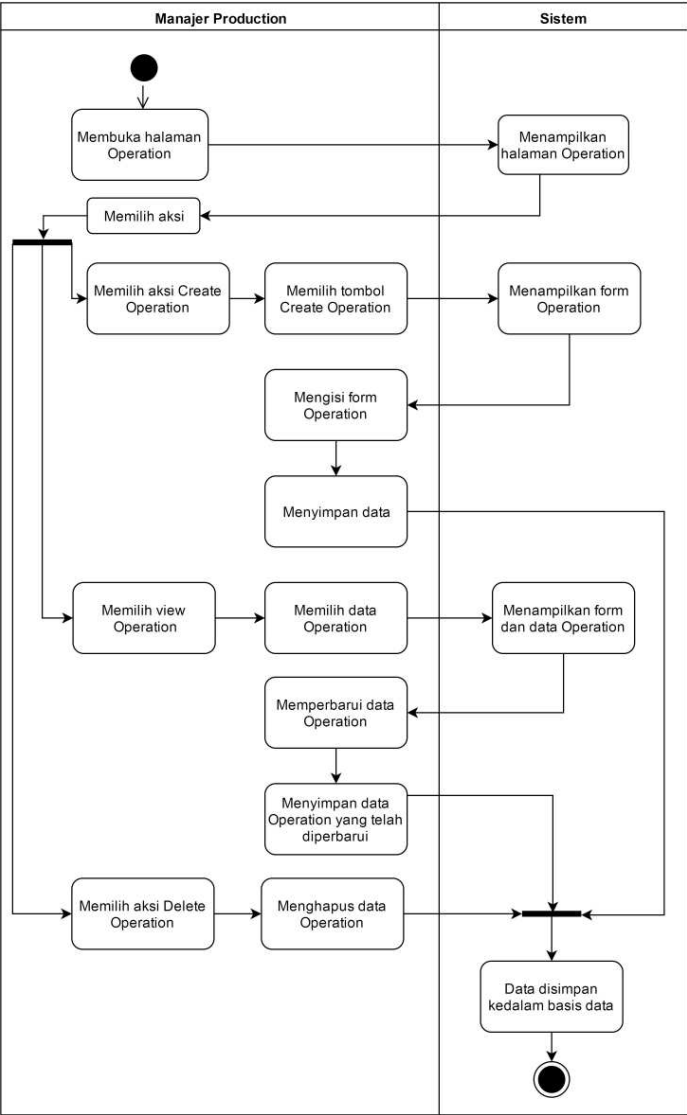


Gambar 3.12 Diagram Aktivitas Mengelola *Routings*

### 3.6.2. Analisis Kasus Penggunaan Mengelola Operasi

Fungsionalitas mengelola data Operasi digunakan sebagai sarana pengelolaan data master Operasi. Pengelolaan data master Operasi hanya bisa dilakukan oleh manajer produksi. Alur aktivitas kasus penggunaan mengelola Operasi ditunjukkan pada Gambar 3.12.

<b>Nama</b>	Mengelola <i>Operasi</i>
<b>Kode</b>	F-002
<b>Deskripsi</b>	Mengelola data master Operasi
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Manajer produksi membuka halaman pengelolaan data Operasi.
<b>Aktor</b>	Manajer dan staff produksi
<b>Aliran:</b>	
- <b>Kejadian Normal</b>	<ol style="list-style-type: none"> <li>1. Manajer produksi membuka halaman pengelolaan data Operasi.</li> <li>2. Sistem menampilkan halaman daftar Operasi.</li> <li>3. Manajer produksi memilih pilihan tambah, ubah, atau hapus pada data Operasi.</li> <li>4. Manajer produksi melakukan penambahan, perubahan, atau penghapusan data.</li> </ol>
- <b>Kejadian Alternatif</b>	Tidak ada
<b>Kondisi Akhir</b>	Sistem menampilkan daftar data Operasi.
<b>Kebutuhan Khusus</b>	Tidak ada

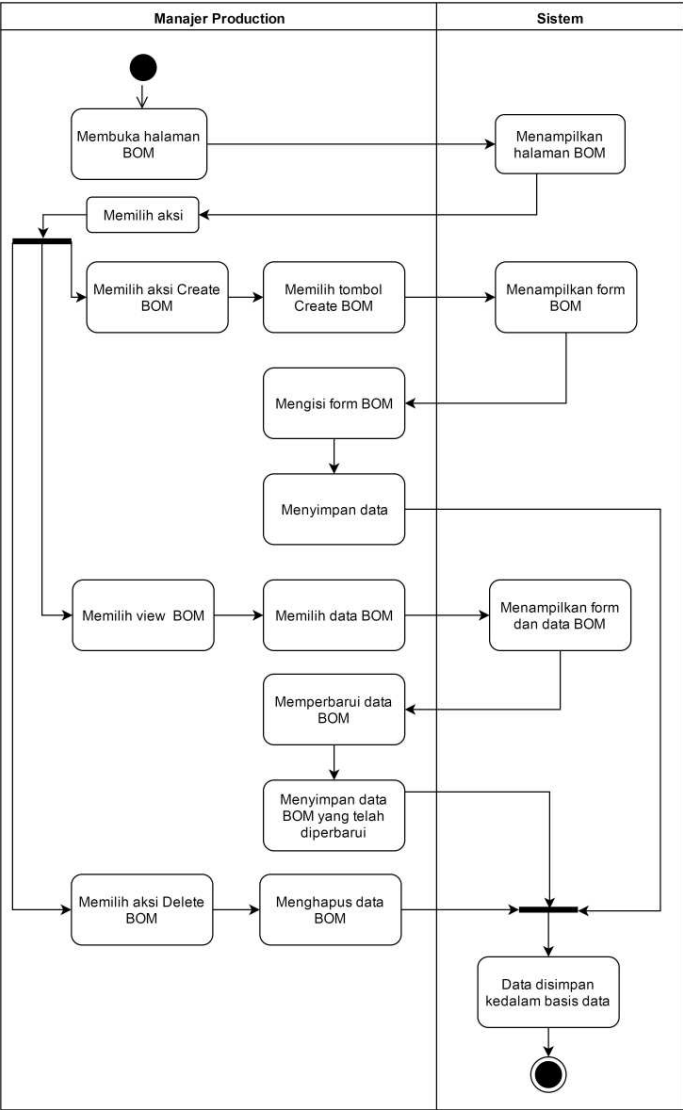


Gambar 3.13 Diagram Aktivitas Mengelola Operasi

### 3.6.3. Analisis Kasus Penggunaan Mengelola *Bill of Material*

Fungsionalitas mengelola *Bill of Material* digunakan sebagai sarana pengelolaan data master *Bill of Material*. Pengelolaan data master *Bill of Material* hanya bisa dilakukan oleh manajer produksi. Alur aktivitas kasus penggunaan mengelola *Bill of Material* ditunjukkan pada gambar 3.13.

<b>Nama</b>	Mengelola <i>Bill of Material</i>
<b>Kode</b>	F-003
<b>Deskripsi</b>	Mengelola data master <i>Bill of Material</i>
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Manajer produksi membuka halaman pengelolaan <i>Bill of Material</i> .
<b>Aktor</b>	Manajer dan staff produksi
<b>Aliran:</b>	
- <b>Kejadian Normal</b>	<ol style="list-style-type: none"> <li>1. Manajer produksi membuka halaman pengelolaan <i>Bill of Material</i>.</li> <li>2. Sistem menampilkan halaman daftar <i>Bill of Material</i>.</li> <li>3. Manajer produksi memilih pilihan tambah, ubah, atau hapus pada data <i>Bill of Material</i>.</li> <li>4. Manajer produksi melakukan penambahan, perubahan, atau penghapusan data.</li> </ol>
- <b>Kejadian Alternatif</b>	Tidak ada
<b>Kondisi Akhir</b>	Sistem menampilkan daftar data <i>Bill of Material</i> .
<b>Kebutuhan Khusus</b>	Tidak ada



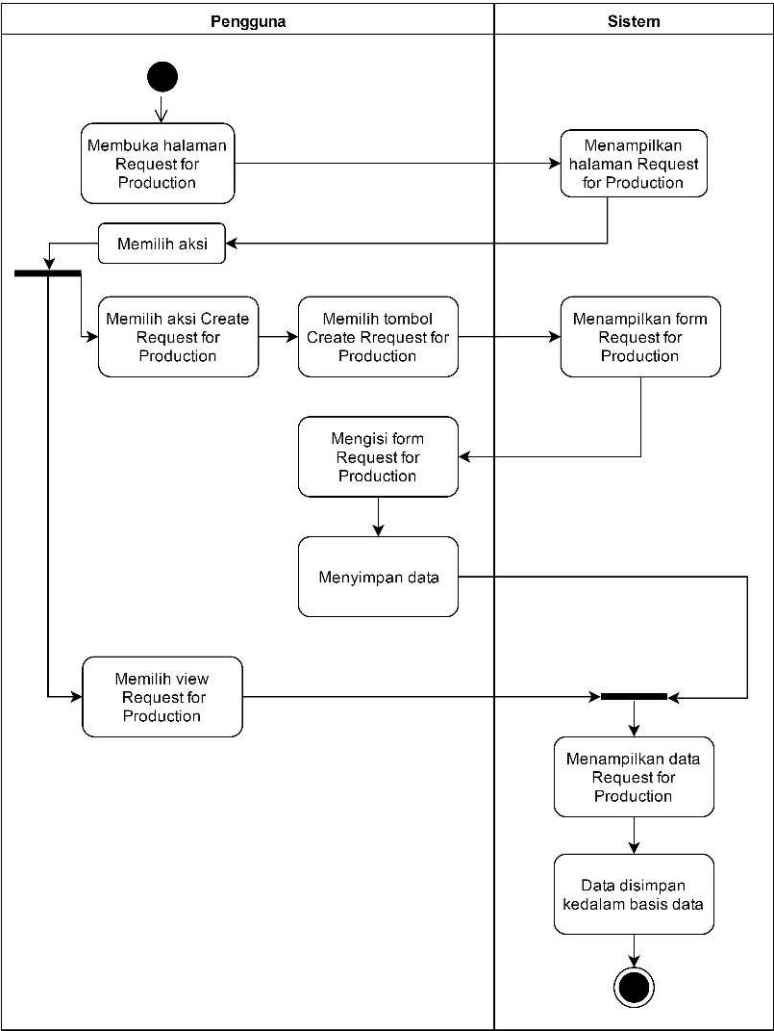
Gambar 3.14 Diagram Aktivitas Mengelola *Bill of Material*



### 3.6.4. Analisis Kasus Penggunaan Membuat *Request for Production*

Fungsionalitas mengelola *Request for Production* digunakan sebagai sarana menambahkan data *Request for Production*. Pengelolaan data *Request for Production* dapat dilakukan baik oleh manajer maupun staff produksi. Alur aktivitas kasus penggunaan membuat *Request for Production* ditunjukkan pada gambar 3.14.

<b>Nama</b>	Mengelola <i>Request for Production</i>
<b>Kode</b>	F-004
<b>Deskripsi</b>	Menambah data <i>Request for Production</i>
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Pengguna membuka halaman pengelolaan <i>Request for Production</i> .
<b>Aktor</b>	Manajer dan staff produksi
<b>Aliran:</b>	
- <b>Kejadian Normal</b>	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman pengelolaan <i>Request for Production</i>.</li> <li>2. Sistem menampilkan halaman daftar <i>Request for Production</i>.</li> <li>3. Pengguna memilih pilihan tambah data <i>Request for Production</i>.</li> <li>4. Pengguna melakukan penambahan data.</li> </ol>
- <b>Kejadian Alternatif</b>	Tidak ada
<b>Kondisi Akhir</b>	Sistem menampilkan data <i>Request for Production</i> yang baru saja ditambahkan.
<b>Kebutuhan Khusus</b>	Tidak ada



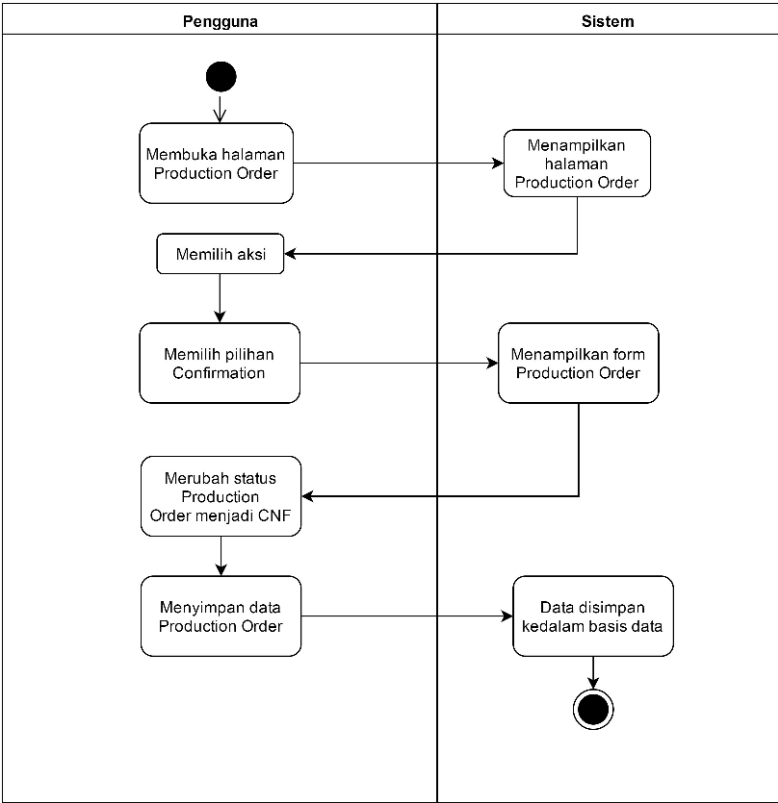
Gambar 3.15 Diagram Aktivitas Membuat *Request for Production*

3.6.5. Analisis Kasus Melakukan *Confirmation*

Fungsionalitas melakukan *Confirmation* digunakan sebagai sarana konfirmasi *Production Order* yang telah selesai

proses produksinya. Melakukan *Confirmation* dapat dilakukan baik oleh manajer maupun staff produksi. Alur aktivitas kasus penggunaan melakukan *Confirmation* ditunjukkan pada gambar 3.15.

<b>Nama</b>	Melakukan <i>Confirmation</i>
<b>Kode</b>	F-005
<b>Deskripsi</b>	Melakukan <i>Confirmation</i> dari data <i>Production Order</i> .
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Pengguna membuka halaman daftar <i>Production Order</i> .
<b>Aktor</b>	Manajer dan staff produksi
<b>Aliran:</b>	
- <b>Kejadian Normal</b>	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman pengelolaan <i>Production Order</i>.</li> <li>2. Sistem menampilkan halaman daftar <i>Production Order</i>.</li> <li>3. Pengguna memilih pilihan <i>Confirmation</i>.</li> <li>4. Pengguna melakukan perubahan data status <i>Production Order</i>.</li> </ol>
- <b>Kejadian Alternatif</b>	Tidak ada
<b>Kondisi Akhir</b>	Sistem menampilkan data <i>Production Order</i> .
<b>Kebutuhan Khusus</b>	Tidak ada



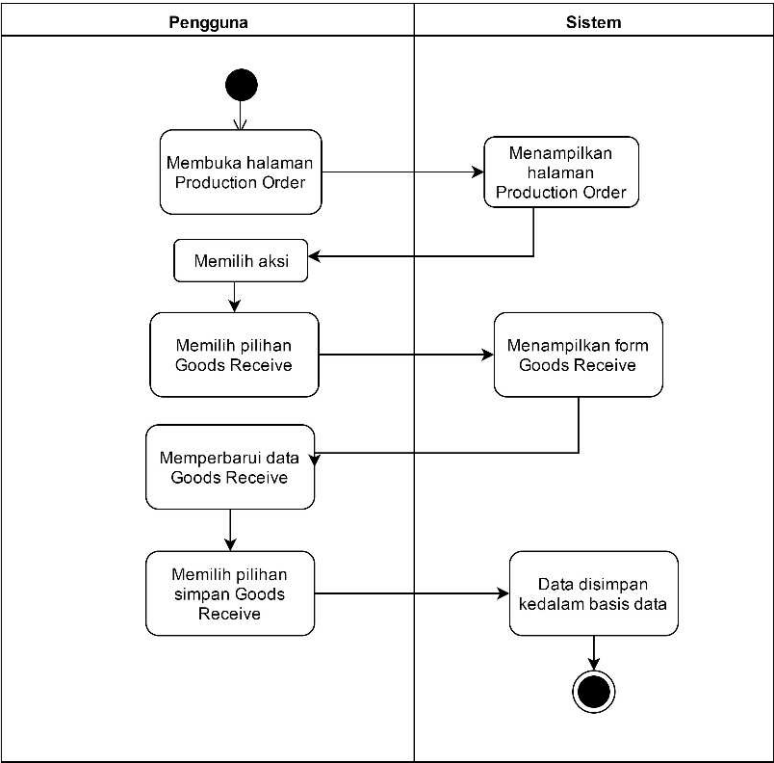
Gambar 3.16 Diagram Aktivitas Melakukan *Confirmation*

3.6.6. Analisis Kasus Penggunaan Membuat *Goods Receive*

Fungsionalitas menambahkan *Goods Receive* digunakan sebagai sarana pencatatan penerimaan barang mentah produksi. Menambah data *Goods Receive* dapat dilakukan baik oleh manajer maupun staff produksi. Alur aktivitas kasus penggunaan membuat *Goods Receive* ditunjukkan pada gambar 3.16.

Nama	Membuat <i>Goods Receive</i>
------	------------------------------

<b>Kode</b>	F-006
<b>Deskripsi</b>	Menambahkan data <i>Goods Receive</i> berupa data penerimaan barang bahan produksi.
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Pengguna membuka halaman daftar <i>Production Order</i> .
<b>Aktor</b>	Manajer dan staff produksi
<b>Aliran:</b>	
- <b>Kejadian Normal</b>	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman pengelolaan <i>Production Order</i>.</li> <li>2. Sistem menampilkan halaman daftar <i>Production Order</i>.</li> <li>3. Pengguna memilih pilihan <i>Goods Receive</i>.</li> <li>4. Pengguna melakukan penambahan data.</li> </ol>
- <b>Kejadian Alternatif</b>	Tidak ada
<b>Kondisi Akhir</b>	Sistem menampilkan data <i>Goods Receive</i> .
<b>Kebutuhan Khusus</b>	Tidak ada



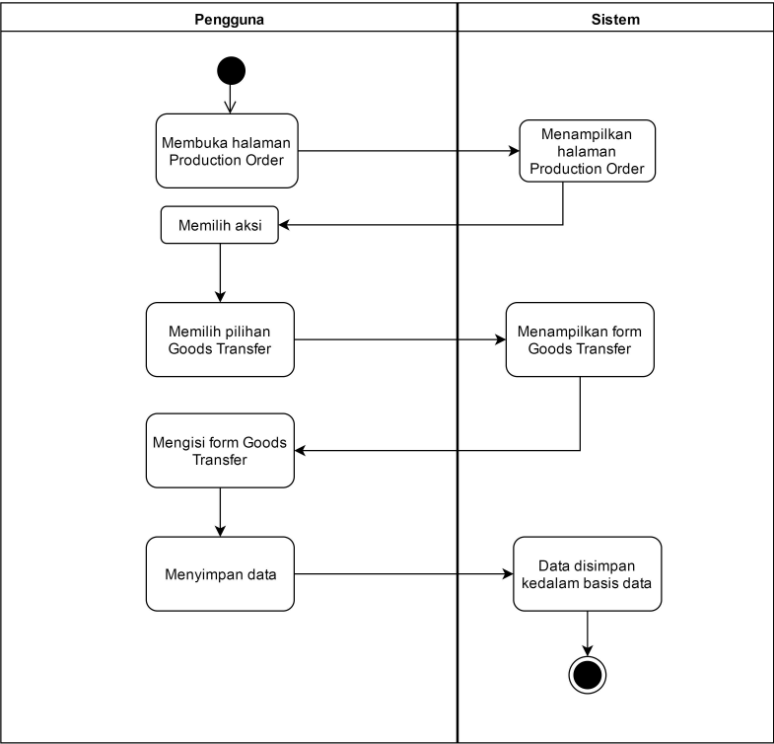
Gambar 3.17 Diagram Aktivitas Membuat *Goods Receive*

3.6.7. Analisis Kasus Penggunaan Membuat *Goods Transfer*

Fungsionalitas menambahkan *Goods Transfer* digunakan sebagai sarana pencatatan pengiriman barang hasil produksi ke Modul Pergudangan. Menambah data *Goods Transfer* dapat dilakukan baik oleh manajer maupun staff produksi. Alur aktivitas kasus penggunaan membuat *Goods Trasnfer* ditunjukkan pada gambar 3.17.

Nama	Membuat <i>Goods Transfer</i>
------	-------------------------------

<b>Kode</b>	F-007
<b>Deskripsi</b>	Menambahkan data <i>Goods Transfer</i> berupa data barang hasil produksi.
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Pengguna membuka halaman daftar <i>Production Order</i> .
<b>Aktor</b>	Manajer dan staff produksi
<b>Aliran:</b>	
- <b>Kejadian Normal</b>	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman pengelolaan <i>Production Order</i>.</li> <li>2. Sistem menampilkan halaman daftar <i>Production Order</i>.</li> <li>3. Pengguna memilih pilihan <i>Goods Transfer</i>.</li> <li>4. Pengguna melakukan penambahan data.</li> </ol>
- <b>Kejadian Alternatif</b>	Tidak ada
<b>Kondisi Akhir</b>	Sistem menampilkan data <i>Goods Transfer</i> .
<b>Kebutuhan Khusus</b>	Tidak ada



Gambar 3.18 Diagram Aktivitas Membuat *Goods Transfer*

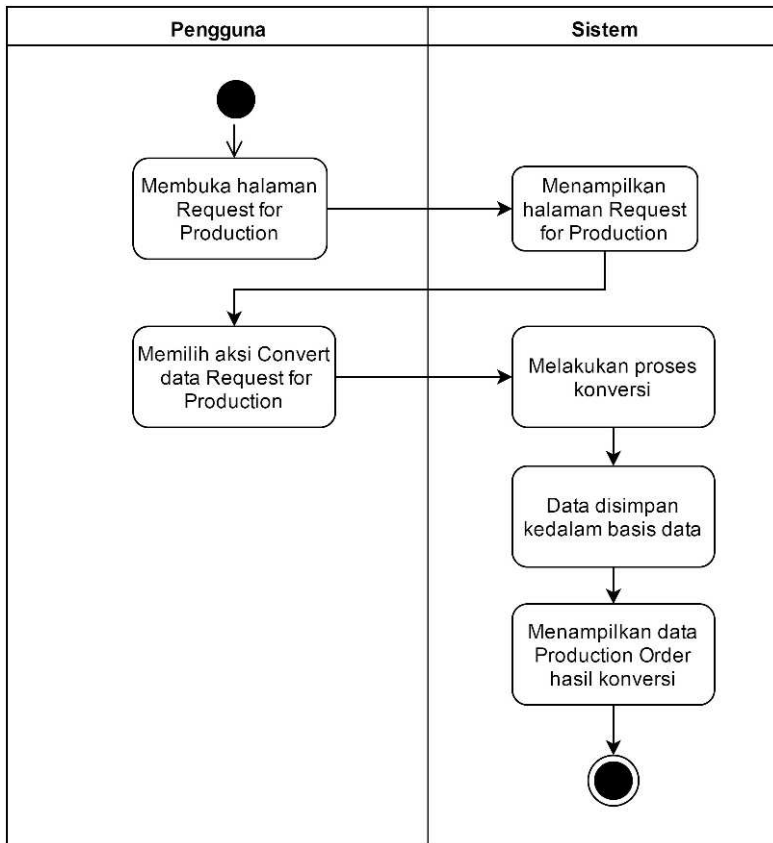
3.6.8. Analisis Kasus Penggunaan *Generating Scheduling*

Fungsionalitas *Generating Scheduling* digunakan sebagai sarana konversi *Request for Production* menjadi *Production Order*. Proses konversi ini merupakan otomatisasi jadwal produksi menggunakan *Genetic Algorithm*. Alur aktivitas kasus penggunaan *Generating Scheduling* ditunjukkan pada gambar 3.18.

Nama	<i>Generating Scheduling</i>
Kode	F-008



<b>Deskripsi</b>	Proses konversi dari <i>Request for Production</i> menjadi <i>Production Order</i> .
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Pengguna membuka halaman daftar <i>Request for Production</i> .
<b>Aktor</b>	Manajer dan staff produksi
<b>Aliran:</b>	
- <b>Kejadian Normal</b>	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman daftar <i>Request for Production</i>.</li> <li>2. Sistem menampilkan halaman daftar <i>Request for Production</i>.</li> <li>5. Pengguna memilih pilihan <i>Convert</i>.</li> <li>6. Sistem melakukan proses konversi <i>Request for Production</i> menjadi <i>Production Order</i>.</li> <li>7. Sistem menampilkan data <i>Production Order</i> yang telah dibuat.</li> </ol>
- <b>Kejadian Alternatif</b>	Tidak ada
<b>Kondisi Akhir</b>	Sistem menampilkan data <i>Production Order</i> .
<b>Kebutuhan Khusus</b>	Tidak ada



**Gambar 3.19** Diagram Aktivitas *Generating Scheduling*

### 3.7. Perancangan *Scheduling*

Pada subbab ini akan dijelaskan tahap-tahap penerapan Algoritma Genetika dalam rangka menyelesaikan permasalahan penjadwalan produksi.

### 3.7.1. Pengkodean Kromosom

Berikut *pseudocode* dari tahap-tahap pengkodean kromosom berdasarkan prosedur Bierwirth pada subbab 2.4.1.

1. Langkah nomor 1 dirancang melalui *pseudocode* pada kode sumber 3.1. Kode sumber 3.1 menjelaskan sebuah *function* untuk memilih operasi-operasi yang mengawali setiap pekerjaan.

```
function getFront(Array of Operations)
  front = Array
  for i = 0 to Operations.length
    push Operations[i] to front
  return front
end function
```

**Kode Sumber 3.1 Pseudocode Membuat Himpunan Operasi Awal**

2.
  - i. Langkah 2.i dirancang melalui *pseudocode* pada kode sumber 3.2. Langkah ini memilih operasi tercepat dari himpunan operasi yang dihasilkan pada langkah 1.

```
function getFastest(Array of Operations)
  index = 0
  for i = 0 to Operations.length
    if (Operations[i][duration] +
        Operations[i][start_time]) <
        (Operations[index][duration] +
        Operations[index][start_time])
      index = i
  return index
end function
```

**Kode Sumber 3.2 Pseudocode Memilih Operasi Tercepat**

- ii. Langkah 2.ii dirancang melalui *pseudocode* pada kode sumber 3.3. Langkah ini memilih operasi-operasi dari langkah 1 yang memiliki sumberdaya yang sama dengan operasi yang terpilih pada langkah 2.i.

```
function getCommon(Array of Operations,
fastestOperation)
  commonOperation = Array
  for = 0 to Operations.length
    if(Operations[i][employee] ==
fastestOperation[employee] and
Operations[i][machine] ==
fastestOperation[machine])
      push Operations[i] to
commonOperation
  return commonOperation
end function
```

**Kode Sumber 3.3 Pseudocode Mencari Operasi dengan Sumberdaya yang Sama**

- iii. Prosedur 2.iii dirancang melalui *pseudocode* pada kode sumber 3.4. Langkah ini memilih operasi tercepat pada operasi-operasi yang terpilih pada langkah 2.ii.

```
function getEarliest(Array of Operations)
  index = 0
  for = 0 to Operations.length
    if(Operations[i][start_time] <
Operations[index][start_time])
      index = i
  return index
end function
```

**Kode Sumber 3.4 Pseudocode Mencari Operasi Paling Awal**

- iv. Prosedur 2.iv dirancang melalui *pseudocode* pada kode sumber 3.5. Langkah ini menghapus operasi yang terpilih pada langkah 2.iii dari himpunan operasi pada langkah 1.

```
function deleteOnA(Array of A, earliestOfB)
  for = 0 to A.length
    if(A[i][start_time] >
      earliestOfB[start_time])
      delete A[i] from A
  return A
end function
```

**Kode Sumber 3.5 Pseudocode Hapus Operasi Tercepat B dari A**

- 3 Prosedur nomor 3 dirancang melalui *pseudocode* pada kode sumber 3.6. Langkah ini melakukan pembaharuan waktu mulai operasi yang terpilih pada langkah 2.iii untuk dirangkai menjadi solusi.

```
function updateTime(Array of Jobs, selectedB)
  selectedB[start_time] =
    max(Jobs[i][start_time]+Jobs[i][duration], Jobs[i-1][start_time]+Jobs[i][duration])
  return selectedB
end function
```

**Kode Sumber 3.6 Pseudocode Memperbarui Waktu Operasi dari Jobs yang Terpilih dari B**

- 4 Prosedur nomor 4 dirancang melalui *pseudocode* pada kode sumber 3.7. Langkah ini melakukan pembaharuan himpunan operasi pada langkah 1 jika terdapat operasi suksesor pada pekerjaan tertentu.

```

function addToA(Array of A, Array of Operations
selectedB)
    if Operations[selectedB[nth_operation]+1] is
    not null
        push
        Operations[selectedB[nth_operation]+1]
        to A
    end function

```

**Kode Sumber 3.7 Pseudocode Memasukan Operasi Selanjutnya**

- 5 Prosedur nomor 5 dirancang melalui *pseudocode* pada kode sumber 3.8. Langkah ini memastikan bahwa seluruh operasi dari pekerjaan yang terlibat telah seluruhnya diproses.

```

function Iterate(Array of Operations)
    Jobs = Array
    front = getFront(Operations)
    while front is not empty
        fastest = getFastest(Operations)
        common = getCommon(Operations, fastest)
        earliest = getEarliest(common)
        deleteOnA(front, earliest)
        updateTime(Jobs, earliest)
        addToA(front, Operations, earliest)
        push earliest to Jobs
    return Jobs
end function

```

**Kode Sumber 3.8 Pseudocode Ulangi Semua Langkah Hingga A Habis**

### 3.7.2. Pindah Silang

Prosedur pindah silang pada subbab 2.4.2 dirancang melalui *pseudocode* pada kode sumber 3.9. Perancangan pindah silang dilakukan dengan melakukan pemilihan *gen-gen* pada

kromosom induk yang terlibat berdasarkan bilangan random sehingga menghasilkan kromosom baru.

```
function crossover(parent1, parent2)
  child = Array
  while parent1 is not null and parent2 is not
    null
    randomized = random(1,2)
    if randomized is 1
      push parent1.top to child
    else if randomized is 2
      push parent2.top to child
  end function
```

**Kode Sumber 3.9 Pseudocode Pindah Silang**

### 3.7.3. Mutasi

Prosedur mutasi pada subbab 2.4.3 dirancang melalui *pseudocode* kode sumber 3.10. Mutasi dilakukan dengan menukar 2 *gen* pada kromosom dengan posisi acak.

```
function mutation(chromosome)
  position1 = random(1,chromosome.length)
  position2 = random(1,chromosome.length)
  Swap(chromosome[position1],chromosome[position
    2])
  end function
```

**Kode Sumber 3.10 Pseudocode Perancangan Mutasi Algoritma Genetika**

### 3.7.4. Fitness Function

Prosedur perhitungan *fitness value* pada subbab 2.4.4 dirancang melalui *pseudocode* kode sumber 3.11. Perhitungan dilakukan dengan membagi 1 dengan durasi pekerjaan.

```

function getFitness(chromosome)
    return 1 /
        chromosome[chromosome.length][duration]
end function

```

**Kode Sumber 3.11 Pseudocode *Fitness Function***

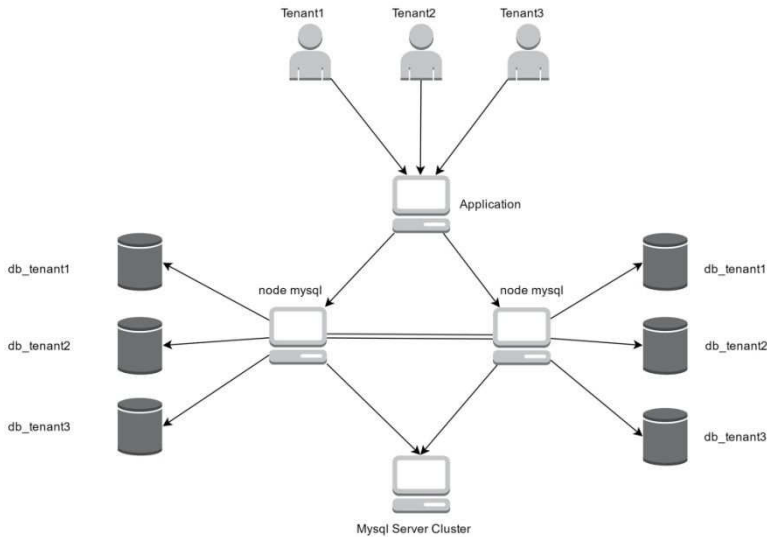
### **3.8. Perancangan Basis Data Terdistribusi**

Perangkat lunak yang akan dibangun menggunakan basis data terdistribusi. Jika terdapat 3 perusahaan yaitu perusahaan 1, perusahaan 2, dan perusahaan 3 maka seluruh perusahaan dapat menggunakan ERP secara bersamaan. Setiap perusahaan mempunyai 2 database yang telah direplikasi yaitu *Computer Server 1* dengan IP Enterprise 1, 10.151.64.182 dan *Computer Server 2* dengan IP Enterprise 2, 10.151.64.203. komputer 3 berfungsi sebagai server cluster untuk melakukan replikasi, sehingga apabila database 1 atau *Computer Server 1* dalam keadaan mati, sistem ERP masih tetap berjalan.

### **3.9. Perancangan Multitenancy**

Pada bagian ini akan dijelaskan mengenai perancangan multi-tenancy yang terdapat pada sistem ERP 2016. Secara rinci mengenai multi-tenancy dijabarkan pada gambar 3.20.





**Gambar 3.20 Perancangan *Multi-tenancy***

Gambar diatas menerangkan bagaimana alur multi-tenancy yang terjadi pada sistem ERP. Pembagian *server* data mysql sudah dijelaskan pada subbab 3.1.2 yaitu dengan cara clustering menjadi 2 *node* yang berbeda. Basis data terdistribusi merupakan penunjang dari *multi-tenancy*, yang berfungsi untuk mengatur session setiap perusahaan yang akan mengakses sistem ERP.

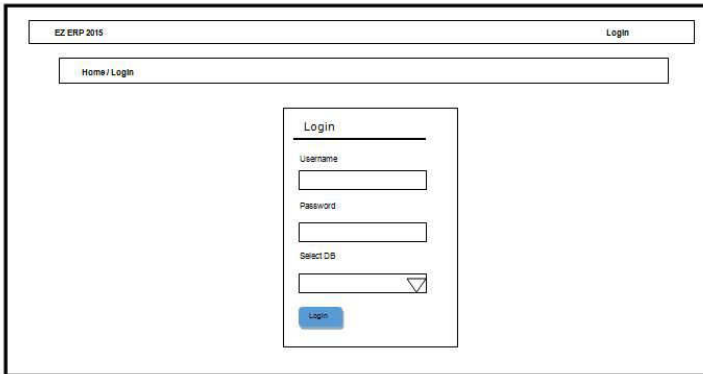
### **3.10. Perancangan RBAC**

Pada bagian ini akan dijelaskan mengenai perancangan RBAC yang terdapat pada sistem ERP 2016. Secara rinci mengenai RBAC dijabarkan sebagai berikut

#### **3.10.1. Perancangan Antarmuka *Login***

Antarmuka *login* digunakan sebagai *form* pengisian data *username*, *password*, dan *tenant* untuk melakukan otentikasi dan

otorisasi pengguna pada aplikasi ERP. Antarmuka *login* digambarkan seperti pada Gambar 3.21.

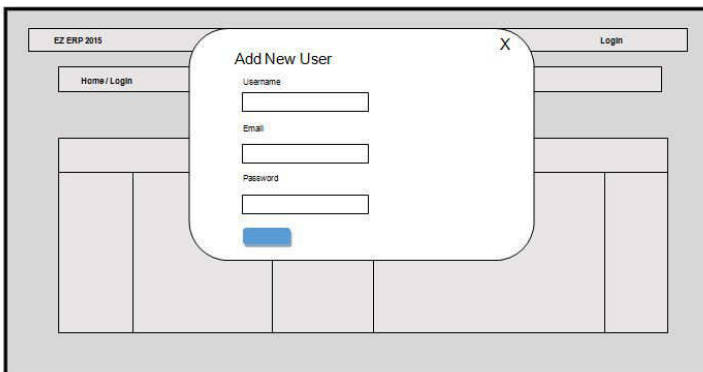


The screenshot shows the login page of the EZ ERP 2015 application. At the top, there is a header bar with "EZ ERP 2015" on the left and "Login" on the right. Below the header, there is a navigation bar with a "Home / Login" link. The main content area features a "Login" form with the following fields: "Username" (text input), "Password" (text input), and "Select DB" (dropdown menu). A blue "Login" button is positioned at the bottom of the form.

**Gambar 3.21 Antarmuka *Login***

### 3.10.2. Perancangan Antarmuka *Add User*

Antarmuka *Add User* digunakan untuk melakukan pendaftaran pengguna baru pada sistem ERP. Antarmuka *Add User* digambarkan pada gambar 3.22.



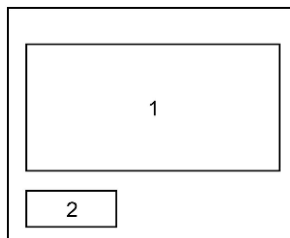
The screenshot shows the "Add New User" dialog box in the EZ ERP 2015 application. The dialog box is centered on the screen and has a title bar with "Add New User" and a close button (X). It contains three input fields: "Username", "Email", and "Password". A blue button is located at the bottom of the dialog box. The background of the application window is dimmed, showing the same header and navigation bar as in the previous screenshot.

**Gambar 3.22 Antarmuka *Add User***

### 3.11. Perancangan Antarmuka Sistem

Pada sub-bab ini membahas perancangan antarmuka yang akan digunakan dalam aplikasi ERP khususnya modul Produksi. Ilustrasi perancangan ditunjukkan sebagai gambar disertai keterangan objek-objek yang ada di dalamnya.

Terdapat beberapa rancangan antarmuka yang akan dijadikan *template*. *Template-template* tersebut digunakan sesuai dengan tingkat kerumitan *form*. Gambar 3.23 menunjukkan *template* rancangan antarmuka yang sederhana dengan bagian nomor 1 adalah posisi *form* dan nomor 2 adalah posisi tombol simpan/hapus. Gambar 3.24 menunjukkan *template* rancangan antarmuka dengan tingkat kerumitan *form* yang lebih tinggi dengan bagian nomor 1 adalah posisi *form* utama, nomor 2 adalah posisi *form* detail, dan nomor 3 adalah posisi tombol simpan/hapus. Sedangkan Gambar 3.25 menunjukkan *template* antarmuka berbentuk *list* dengan *input form* yang diwakili oleh angka-angka.



**Gambar 3.23 *Template* Perancangan Antarmuka Sederhana**

**Gambar 3.24 *Template* Perancangan Antarmuka Rumit**

**Gambar 3.25 *Template* Perancangan Antarmuka *list***

### **3.11.1. Perancangan Antarmuka Halaman *Input Data Routings***

Pada halaman ini, pengguna dapat memasukkan, serta merubah data *Routings*. Rancangan antarmuka halaman *input data Routings* menggunakan *template* pada Gambar 3.23.

Berikut penjelasan bagian-bagian antarmuka form data *Routings*:

1. *Dropdown List* dari barang yang akan diproduksi.
2. *Dropdown List* dari operasi-operasi penyusun *Routing*.
3. Tombol hapus operasi.

4. Tombol simpan data *Routing*.

### **3.11.2. Perancangan Antarmuka Halaman *Input Data Operasi***

Pada halaman ini, pengguna dapat memasukkan, merubah, serta menghapus data Operasi. Rancangan antarmuka halaman *input data Operasi* menggunakan *template* pada Gambar 3.23.

Berikut penjelasan bagian-bagian antarmuka form data Operasi:

1. Nama Operasi.
2. *Dropdown List* dari tipe Operasi.
3. Tombol simpan data Operasi.

### **3.11.3. Perancangan Antarmuka Halaman *Input Data Bill of Material***

Pada halaman ini, pengguna dapat memasukkan, merubah, serta menghapus data *Bill of Material*. Rancangan antarmuka halaman *input data Bill of Material* menggunakan *template* pada gambar 3.24.

Berikut penjelasan bagian-bagian antarmuka form data *Bill of Material*:

1. *Dropdown List* nama barang utama.
2. *Input* data deskripsi singkat *Bill of Material*.
3. *Dropdown List* dari nama material-material penyusun.
4. *Input* data kuantitas material-material penyusun.
5. Tombol hapus data material.
6. Tombol tambah data material.
7. Tombol simpan data *Bill of Material*.

### **3.11.4. Perancangan Antarmuka Halaman *Input Data Request for Production***

Pada halaman ini, pengguna dapat memasukkan data *Reqeust for Production*. Rancangan antarmuka halaman *input*

data *Request for Production* menggunakan *template* pada gambar 3.23.

Berikut penjelasan bagian-bagian antarmuka form data *Request for Production*:

1. *Dropdown List* nama barang yang akan diproduksi.
2. *Input* data kuantitas barang yang akan diproduksi..
3. *Input* data tanggal mulai produksi.
4. *Input* data tanggal selesai produksi.
7. Tombol simpan data *Request for Production*.

### **3.11.5. Perancangan Antarmuka Halaman *Goods Receive***

Pada halaman ini, pengguna dapat memasukkan data *Goods Receive*. Rancangan antarmuka halaman *Goods Receive* menggunakan *template* pada gambar 3.24.

Berikut penjelasan bagian-bagian antarmuka form *input* data *Goods Receive* :

1. *Dropdown List* yang berisi pilihan *Production Order ID*.
2. *Input* data tanggal *Goods Receive*.
3. *Dropdown List* dari nama-nama barang di dalam *Goods Receive*.
4. *Input* data jumlah barang yang direncanakan.
5. *Input* data jumlah barang yang sebenarnya.
6. Tombol hapus data barang.
7. Tombol tambah data barang.
8. Tombol simpan data *Goods Receive*.

### **3.11.6. Perancangan Antarmuka Halaman *Confirmation***

Pada halaman ini pengguna dapat melakukan konfirmasi *Production Order*. Rancangan antarmuka halaman *Confirmation* menggunakan *template* pada gambar 3.23.

Berikut penjelasan bagian-bagian antarmuka form *input* data *Production Order* untuk melakukan konfirmasi :

1. *Dropdown List* berisi pilihan ID *Production Request*.
2. *Dropdown List* berisi pilihan nama barang yang akan diproduksi.
3. *Input* data jumlah barang yang akan diproduksi.
4. *Input* data tanggal produksi.
5. *Input* data tanggal selesai produksi.
6. *Input* data jumlah barang yang sudah diproduksi.
7. *Dropdown List* berisi pilihan *Production Order status*.

### **3.11.7. Perancangan Antarmuka Halaman *Goods Transfer***

Pada halaman ini pengguna dapat memasukkan data *Goods Transfer*. Rancangan antarmuka halaman *Goods Transfer* menggunakan *template* pada gambar 3.23.

Berikut penjelasan bagian-bagian antarmuka form *input* data *Goods Transfer*:

1. *Dropdown List* yang berisi pilihan *Production Order ID*.
2. *Dropdown List* yang berisi barang hasil produksi.
3. *Input* data jumlah barang hasil produksi.
4. *Input* data tanggal *Goods Transfer*.
5. Tombol simpan data *Goods Transfer*.

### **3.11.8. Perancangan Antarmuka Halaman *Generating Scheduling***

Pada halaman ini, pengguna dapat melakukan konversi *Request for Production* menjadi *Production Order*. Rancangan

antarmuka halaman *Generating Scheduling* digambarkan pada gambar 3.25.

Berikut penjelasan bagian-bagian antarmuka halaman *Generating Scheduling*:

1. *Static Input* nama barang.
2. *Static Input* jumlah barang.
3. *Static Input* tanggal produksi.
4. *Static Input* tanggal selesai produksi.
5. *Button* pilihan *Convert*.



## **BAB IV IMPLEMENTASI SISTEM**

Bab ini membahas implementasi dari perancangan sistem. Di dalamnya mencakup penjelasan lingkungan pengembangan sistem serta proses implementasi *distributed database*, *RBAC*, *multi-tenancy*, dan antarmuka pengguna.

### **4.1. Lingkungan Pengembangan Sistem**

Lingkungan pengembangan sistem yang digunakan untuk mengembangkan Tugas Akhir ini dilakukan pada lingkungan dan kaskas sebagai berikut.

1. Database yang digunakan pada server adalah MySQL Cluster.
2. PC untuk server menggunakan AMD A6-3400M APU with Radeon HD Graphics @1.40Ghz, RAM 4GB dengan Sistem Operasi Windows 7 Ultimate.  
StarUML untuk pembuatan diagram, Power Designer 12.5, Sublime sebagai teks editor, draw.io untuk pembuatan desain antarmuka, Power Designer untuk pembuatan CDM dan PDM.

Mozilla Firefox 46.0.1 sebagai antarmuka untuk pengujian aplikasi klien.

### **4.2. Implementasi Basis Data Terdistribusi**

Pada bagian ini akan dijelaskan mengenai implementasi *distributed database* yang terdapat pada sistem ERP 2016. Secara rinci mengenai implementasi *distributed database* dijabarkan sebagai berikut:

#### **4.2.1. Instalasi Data dan SQL node pada node1 dan node2**

Pada implementasi instalasi dan sql node pada node1 dan node2, dilakukan langkah-langkah sebagai berikut:

1. Membuat grup MySQL pengguna baru, kemudian menambah user MySQL. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.1.

```
shell> groupadd mysql
shell> useradd-g mysql mysql
```

**Kode Sumber 4.1 Membuat grup MySQL pengguna baru dan menambah user MySQL**

2. Mengubah lokasi ke dalam direktori yang berisi file yang telah didownload, kemudian mengubah arsip dan menciptakan symlink ke dalam direktori mysql yang bernama “mysql”. Hal yang perlu diperhatikan adalah, file yang sebenarnya dan nama direktori bervariasi sesuai dengan jumlah cluster versi MySQL. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.2.

```
shell> cd/usr/local
shell>/usr/local$ tar xzvf mysql-cluster-gpl-
7.1.34-linux-x86_64-glibc23
shell>ln-s/usr/local/mysql-cluster-gpl-7.1.34-
linux-x86-glibc23/usr/local/mysql
shell> export PATH = $ PATH:/usr/local/mysql/bin
shell> echo "export PATH=¥$PATH:/usr/local/
mysql/bin">> /etc/bash.bashrc
```

**Kode Sumber 4.2 Mengubah lokasi ke dalam direktori**

3. Mengubah lokasi ke dalam direktori mysql dan menjalankan script untuk menciptakan *database system*. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.3.

```
shell> cd mysql
shell>. / scripts / mysql_install_db-user = mysql
```

**Kode Sumber 4.3 Menjalankan script MySQL**

4. Mengatur izin yang diperlukan oleh server MSQL. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.4.

```
shell> chown-R root.
shell> chown-R mysql data
shell> chgrp-R mysql.
```

**Kode Sumber 4.4 Mengatur izin yang diperlukan oleh server MSQL**

5. Menyalin *script startup MySQL* ke direktori yang sesuai, mengubah menjadi executable, dan memulai ketika sistem beroperasi. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.5.

```
shell> cp support-files/mysql.server / etc / init.d / mysql
shell> chmod + x / etc / init.d / mysql
shell> update-rc.d mysql defaults
```

**Kode Sumber 4.5 Menyalin script startup MySQL**

#### 4.2.2. Pemasangan Node Management pada node03

Pemasangan node manajemen memerlukan manajemen server MySQL Cluster (ndb\_mgmd), diasumsikan bahwa mysql-cluster-gpl-7.1.5-linux-i686-glibc23.tar.gz telah ditempatkan di /var / tmp. Untuk memasang ndb\_mgmd dan ndb\_mgm pada host Cluster, sistem sebagai root melakukan langkah-langkah sebagai berikut:

1. Mengubah lokasi ke dalam direktori / var / tmp direktori, dan mengekstrak ndb\_mgm dan ndb\_mgmd dari arsip ke direktori yang sesuai seperti / usr / local / bin. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.6.

```
shell> cd/usr/local
shell> tar-zxvf mysql-cluster-gpl-7.1.34-linux-x86-glibc23.tar.gz
shell> cd / usr/local/mysql-cluster-gpl-7.1.34-
```

```
linux-x86-glibc235
shell> cp bin / ndb_mgm */usr/local/bin
```

#### **Kode Sumber 4.6 Ekstraksi ndb\_mgm dan ndb\_mgmd**

2. Mengubah lokasi ke dalam direktori tempat file disalin, kemudian dieksekusi. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.7.

```
shell> cd/usr/local/bin
shell> chmod +x ndb_mgm*
```

#### **Kode Sumber 4.7 Mengubah Lokasi Direktori**

### **4.2.3. Konfigurasi Manajemen Node**

Konfigurasi pada manajemen node dilakukan dengan langkah-langkah sebagai berikut:

1. Membuat direktori tempat file konfigurasi ditemukan kemudian membuat file itu sendiri. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.8.

```
shell> mkdir / var / lib / mysql-cluster
shell> cd / var / lib / mysql-cluster
vi config.ini
```

#### **Kode Sumber 4.8 Membuat Direktori Tempat File Konfigurasi**

2. Mengatur file “config.ini”. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.9.

```
[ndbd default]
NoOfReplicas=2
DataMemory=10G
IndexMemory=2G
MaxNoOfAttributes=10000
MaxNumberOfTables=2500
MaxOfOrderedIndexes=4086
MaxOfConcurrentOperations=250000
MaxOfConcurrentOperations=250000
[tcp default]
[ndb_mgmd]
hostname=10.151.64.182
datadir=/var/lib/mysql-cluster
[ndbd]
hostname=10.151.64.182
datadir=/usr/local/mysql/data
[ndbd]
hostname=10.151.64.203
datadir=/usr/local/mysql/data
[mysqld]
MaxNoOfAttributes=10000
hostname=10.151.64.182
[mysqld]
MaxNoOfAttributes=10000
hostname=10.151.64.203
```

**Kode Sumber 4.9 Mengatur File “config.ini”**

#### **4.2.4. Konfigurasi Data dan SQL Node**

Konfigurasi data dan SQL Node dilakukan dengan cara mengedit file my.cnf pada direktori /etc/. Untuk setiap data node dan SQL node yang diatur pada my.cnf. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.10.

```
[client]
port=3306
socket=/tmp/mysql.sock
[mysqld]
port=3306
socket=/tmp/mysql.sock
ndbcluster
ndb-connectstring=10.151.64.181
[mysql_cluster]
ndb-connectstring=10.151.64.181
```

**Kode Sumber 4.10 Konfigurasi SQL Node**

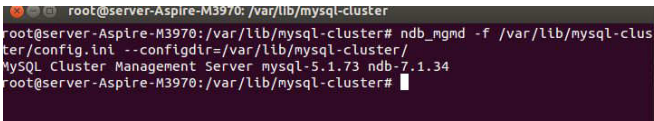
#### 4.2.5. Memulai MySQL Cluster

Setiap proses node cluster harus dimulai secara terpisah. Manajemen node harus dimulai terlebih dahulu, kemudian node data. Pada setiap node SQL dilakukan langkah sebagai berikut :

1. Pada node03 (host manajemen), untuk memulai proses manajemen node dari shell sistem dilakukan perintah berikut. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.11. Jika berhasil akan muncul tampilan yang ditunjukkan oleh Gambar 4.1.

```
shell> ndb_mgmd -f /var/lib/mysql-cluster/config.ini-  
configdir=/var/lib/mysql-cluster/
```

**Kode Sumber 4.11 Memulai Manajemen Node**



**Gambar 4.1 Berhasil Menjalankan Management Node**

2. Jalankan perintah untuk memulai ndbd dan proses mysql server pada masing-masing Data/host SQL. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.12. Jika berhasil akan keluar tampilan seperti Gambar 4.2.

```
shell> /usr/local/mysql/bin/ndbd
```

**Kode Sumber 40.12 Memulai ndbd**

```

root@master-Aspire-M3970: /home/master
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ^C^C -- exit!
Aborted
root@master-Aspire-M3970:/home/master# ndb_mgm
-- NDB Cluster -- Management Client --
ndb_mgm> show
Connected to Management Server at: 10.151.64.181:1186
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
  id=2 @10.151.64.182 (mysql-5.1.73 ndb-7.1.34, Nodegroup: 0)
  id=3 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34, Nodegroup: 0, *)
[ndb_mgmd(MGM)] 1 node(s)
  id=1 @10.151.64.181 (mysql-5.1.73 ndb-7.1.34)
[mysqld(API)] 2 node(s)
  id=4 @10.151.64.182 (mysql-5.1.73 ndb-7.1.34)
  id=5 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34)
ndb_mgm>

```

**Gambar 4.2 Ndbd Berhasil Dijalankan**

3. Mengaktifkan mysql pada data node. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.13. Jika berhasil yang maka akan terlihat tampilan yang ditunjukkan oleh Gambar 4.3.

```
shell> / etc / init.d / mysql start
```

**Kode Sumber 4.13 Memulai MySQL**

```

root@master-Aspire-M3970: /home/master
root@master-Aspire-M3970:/home/master# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.1.73-ndb-7.1.34-cluster-gpl MySQL Cluster Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

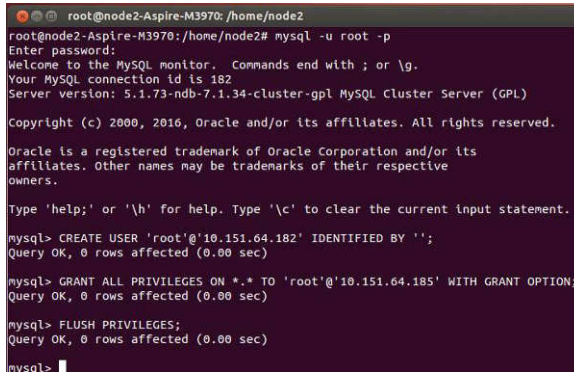
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| db_baru |
| db_baru_1 |
| ezerp |
| mysql |
| ndb_1_fs |
| ndbinfo |
| test |
+-----+
8 rows in set (0.03 sec)

mysql>

```

**Gambar 4.3 MySQL Telah Aktif**

4. Menguji pada client node dengan perintah `ndb_mgm`. Jika berhasil yang maka akan terlihat tampilan yang ditunjukkan oleh Gambar 4.4.



```

root@node2-Aspire-M3970: /home/node2
root@node2-Aspire-M3970:/home/node2# mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 182
Server version: 5.1.73-ndb-7.1.34-cluster-gpl MySQL Cluster Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE USER 'root'@'10.151.64.182' IDENTIFIED BY '';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'10.151.64.185' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql>

```

**Gambar 4.4 Data Node Saling Terkoneksi**

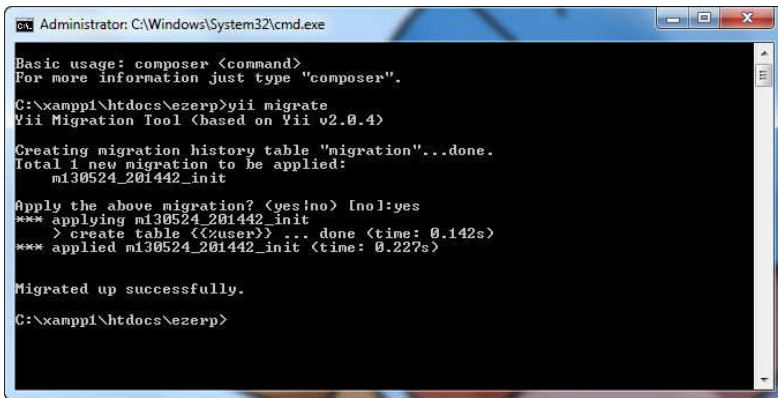
### 4.3. Implementasi RBAC (singkatan)

Pada bagian ini akan dijelaskan mengenai implementasi *RBAC* yang terdapat pada sistem ERP 2016. Secara rinci mengenai implementasi *RBAC* dijabarkan sebagai berikut:

#### 4.3.1. Membuat Tabel Pengguna

Pembuatan tabel pengguna dilakukan dengan menjalankan perintah `yii migrate` pada folder aplikasi. Jika berhasil yang maka akan terlihat tampilan yang ditunjukkan oleh Gambar 4.5.





```

Administrator: C:\Windows\System32\cmd.exe

Basic usage: composer <command>
For more information just type "composer".

C:\xampp1\htdocs\ezerp>yii migrate
Yii Migration Tool (based on Yii v2.0.4)

Creating migration history table "migration"...done.
Total 1 new migration to be applied:
    m130524_201442_init

Apply the above migration? <yes|no> [no]:yes
*** applying m130524_201442_init
    > create table {<%user>} ... done <time: 0.142s>
*** applied m130524_201442_init <time: 0.227s>

Migrated up successfully.

C:\xampp1\htdocs\ezerp>

```

**Gambar 4.5** Proses Migrasi Tabel Pengguna Berhasil

#### 4.3.2. Membuat 4 Tabel Autentifikasi RBAC dan Tabel Pengguna

Pada tahap ini dibutuhkan 4 tabel autentifikasi yang terdiri dari:

1. Tabel Item
2. Tabel Child
3. Tabel Assignment
4. Tabel Rule

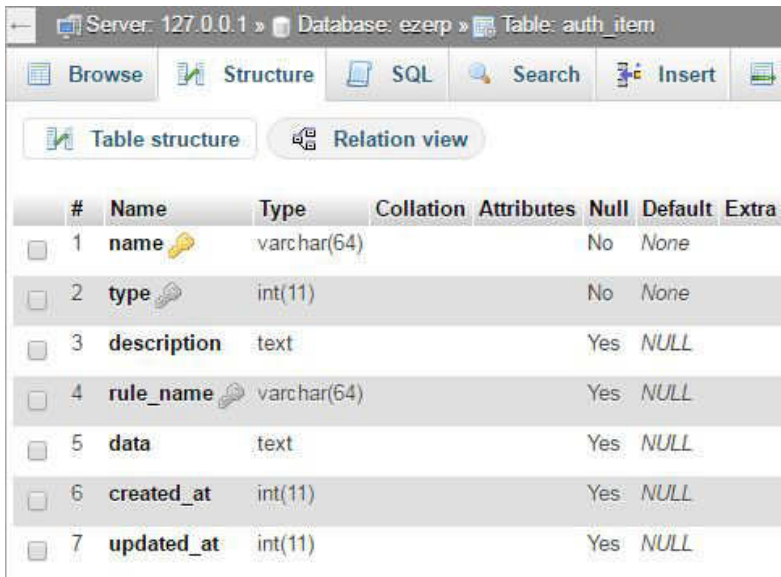
Cara untuk membuat tabel-tabel tersebut yaitu dengan menjalankan perintah yang ditunjukkan pada Kode Sumber 4.14. Jika berhasil yang maka akan terlihat 4 tabel pada phpmyadmin seperti yang ditunjukkan oleh Gambar 4.6.

```

Yii migrate --
migrationPath=@yii/rbac/migrations

```

**Kode Sumber 4.14** *Generate* tabel RBAC



The screenshot shows a database management interface with the following details:

- Server: 127.0.0.1 » Database: ezerp » Table: auth\_item
- Navigation buttons: Browse, Structure, SQL, Search, Insert.
- Active tabs: Table structure, Relation view.
- Table structure details:

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	name	varchar(64)			No	None	
2	type	int(11)			No	None	
3	description	text			Yes	NULL	
4	rule_name	varchar(64)			Yes	NULL	
5	data	text			Yes	NULL	
6	created_at	int(11)			Yes	NULL	
7	updated_at	int(11)			Yes	NULL	

**Gambar 4.6 Proses Migrasi Tabel Pengguna Berhasil**

#### 4.3.3. Membuat Modul Admin

Proses ini bertujuan untuk meletakkan konfigurasi pengguna dan masing-masing model dari 4 tabel autentifikasi. Kemudian dilakukan konfigurasi pada folder config file web.php. Terdapat 3 konfigurasi yaitu:

1. Admin : digunakan sebagai akses ke modul admin.
2. Auth Manager : sebagai autentifikasi di yii2 dan mengatur role default sebagai *guest*.
3. Session Time Out : mengatur durasi time out session selama 5 menit atau 300 detik.

#### 4.3.4. Membuat Model Tabel Autentifikasi, Controller dan View Pengguna

Pada tahap ini dilakukan *generate* kelas model dari masing-masing tabel autentifikasi pada modul admin yang

diperlukan pada tahap sebelumnya. Kemudian ditambahkan generate kelas *controller* dan *view* pada tabel pengguna. Proses generate ini menggunakan *yii generator* yang telah disediakan oleh framework yii.

#### 4.3.5. Menambahkan Kode pada user controller modul admin

Pada modul admin, file *usercontroller.php* ditambahkan kode fungsi untuk semua tabel autentifikasi. Masing-masing fungsi ditunjukkan pada Kode Sumber 4.15.

```
public function actionAuthItem()
{
    $auth = Yii::$app->authManager;

    // menambahkan akses sebagai admin ke tabel
    auth_item
        $admin = $auth->createPermission('admin');
        $admin->description = 'Allow user to
        access all page';
        $auth->add($admin);
}
```

**Kode Sumber 4.15 Fungsi Tabel AuthItem**

### 4.4. Implementasi Multitenancy

Pada subbab ini akan dibahas mengenai implementasi alur proses aplikasi yang telah dirancang pada Bab III. Alur proses aplikasi akan dibahas mulai dari pengambilan data partisipan, hingga proses peningkatan level dan penghentian pada setiap *training*.

#### 4.4.1. Membuat Halaman Muka Tenant

Pada implementasi membuat halaman tenant ini dilakukan pembuatan halaman tenant secara sederhana, kemudian ditambahkan pembuatan database untuk setiap tenant. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.16.

```
// mendapatkan nilai yang dimasukkan dari view
```

```

$host = "10.151.64.182";
$tenant = Yii::$app->request->post('tenant');
$database = Yii::$app->request->post('database');

// membuka koneksi pada untuk memasukkan data tenant
$koneksidb = mysqli_connect($host, "root", "",
    "multitenant");
if ($koneksidb->connect_error) {
    die("Connection failed: " . $koneksidb->connect_error);
}
// query untuk memasukkan data tenant
$insert = "insert into tenant
(tenant,database_name,created_at)
('".$tenant."','".$database."',NOW())";

if($tenant != "" && $database != ""){
    $koneksidb->query($insert);
}
// memutuskan koneksi ke server
$koneksidb->close();

```

**Kode Sumber 4.16 Pembuatan Halaman Muka Tenant**

#### 4.4.2. Menambah *Database* untuk Tenant Baru

Pada tahap ini dilakukan penambahan database untuk tenant baru dengan cara melakukan konfigurasi pada server node mysql. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.17.

```

'mysql -u root -p;' lalu tekan enter
'create database <nama database>;' lalu tekan enter
'use <nama database>;' lalu tekan enter
e /tmp/mysql-dump/final-db.sql

```

**Kode Sumber 4.17 Penambahan Database Tenant Baru**

#### 4.4.3. *Login* Tenant

Setelah proses pembuatan database dan replikasi pada tahap 4.4.2. selesai, maka tenant melakukan login dengan memilih salah satu database, nama database yang dipilih tersebut disimpan dalam session dan akan digunakan untuk koneksi yang akan dibuat. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.18.

```

$host = '10.151.64.182'; // node mysql

```

```

        $port = 3306;
        $waitTimeoutInSeconds = 1;
        $src =
            @fsockopen($host,$port,$errCode,$errStr,$waitTime
            outInSeconds);
        if(is_resource($src)){
            $_SESSION['dbserver_ip'] = "10.151.64.182"; // node
            mysql
            // penyimpanan nama database sebagai session untuk
            multitenancy
            $dbname = isset($_SESSION['database_name']) ?
            $_SESSION['database_name'] : 'test';
        } else {
            $_SESSION['dbserver_ip'] = "10.151.64.203"; // node
            mysql
        }

        $connection = [
            'class' => 'yii\db\Connection',
            'dsn' =>
            'mysql:host=.'.$_SESSION['dbserver_ip'].';dbname='
            .$dbname.'',
            'username' => 'root',
            'password' => '',
            'charset' => 'utf8'
        ];
        return $connection;

```

**Kode Sumber 4.18 Login Tenant**

#### 4.5. Implementasi Antarmuka Modul Produksi

Pada bagian ini akan dijelaskan mengenai implementasi antarmuka yang terdapat pada Modul Produksi. Implementasi antarmuka Modul Produksi mencakup hal-hal berikut:

1. Formulir *input* data *Routings*.
2. Formulir *input* data Operasi.
3. Formulir *input* data *Bill of Material*.
4. Formulir *input* data *Request for Production*.
5. Formulir *Confirmation* data *Production Order*.
6. Formulir *input* data *Goods Receive*.
7. Formulir *input* data *Goods Transfer*.
8. Halaman *Generating Scheduling*.

#### 4.5.1. Implementasi Antarmuka Formulir *Input* data *Routing*s

Halaman antarmuka *input* data *Routing* ditunjukkan oleh Gambar 4.7. Berikut penjelasan bagian-bagian antarmuka formulir *input* data *Routing*s:

1. *Dropdown List* dari barang yang akan diproduksi.
2. *Dropdown List* dari operasi-operasi penyusun *Routing*.
3. Tombol hapus operasi.
4. Tombol tambah data Operasi.
5. Tombol simpan data *Routing*.

Home / Pp Routing / Create Pp Routing

### Create Pp Routing

Material Produced **1**

Choose Pp material

#	Pp operation	
1	Choose Pp operation <b>2</b>	🗑️ <b>3</b>
2	Choose Pp operation	🗑️
3	Choose Pp operation	🗑️

**+Add Row** **4**

**Create** **5**

**Gambar 4.7 Implementasi Antarmuka *Input* Data *Routing*s**

#### 4.5.2. Implementasi Antarmuka Formulir *Input* data *Operasi*

Halaman antarmuka *input* data Operasi ditunjukkan oleh gambar 4.8. Berikut penjelasan bagian-bagian antarmuka formulir *input* data Operasi:

1. Nama Operasi.

2. *Dropdown List* dari tipe Operasi.
3. Tombol simpan data Operasi.

Home / Pp Operation / Create Pp Operation

## Create Pp Operation

Operation **1**

Type **2**

Create **3**

**Gambar 4.8 Implementasi Antarmuka *Input Data Operasi***

### **4.5.3. Implementasi Antarmuka Formulir *Input data Bill of Material***

Halaman antarmuka *input data Bill of Material* ditunjukkan oleh gambar 4.9. Berikut penjelasan bagian-bagian antarmuka formulir *input data Bill of Material*:

1. *Dropdown List* nama barang utama.
2. *Input* data deskripsi singkat *Bill of Material*..
3. *Dropdown List* dari nama material-material penyusun.
4. *Input* data kuantitas material-material penyusun.
5. Tombol hapus data material.
6. Tombol tambah data material.
7. Tombol simpan data *Bill of Material*.

Home / Pp Bom / Create Pp Bom

Create Pp Bom

Material ID 1

Choose lwm item master

Description 2

Description

Pp Bom Material

Showing 1-2 of 2 items.

#	lwm item master	Quantity	
1	<div>Choose lwm item master</div>	<div></div>	<div></div> 5
2	<div>Choose lwm item master</div>	<div></div>	<div></div>

+Add Row 6

Create 7

Gambar 4.9 Implementasi Antarmuka *Input Data Bill of Material*

4.5.4. Implementasi Antarmuka Formulir *Input data Request for Production*

Halaman antarmuka *input data Request for Production* ditunjukkan oleh gambar 4.10. Berikut penjelasan bagian-bagian antarmuka formulir *input data Request for Production*:

- 1. *Dropdown List* nama barang yang akan diproduksi.
- 2. *Input* data kuantitas barang yang akan diproduksi..
- 3. *Input* data tanggal mulai produksi.
- 4. *Input* data tanggal selesai produksi.
- 5. Tombol simpan data *Request for Production*.



Home / Pp Production Request / Create Pp Production Request

## Create Pp Production Request

**Material Produced** 1

Choose Item / Item master

**Quantity** 2

Quantity

**Production Date** 3

Choose Production Date

**Finish Date** 4

Choose Finish Date

**Create** 5

**Gambar 4.10 Implementasi Antarmuka Input Data *Request for Production***

#### 4.5.5. Implementasi Antarmuka Formulir *Confirmation data Production Order*

Halaman antarmuka *input* data *Production Order* ditunjukkan oleh gambar 4.11. Berikut penjelasan bagian-bagian antarmuka *input* data *Production Order*:

1. *Dropdown List* dari *Request for Production ID*.
2. *Dropdown List* dari nama barang yang diproduksi..
3. Kuantitas barang yang diproduksi.
4. *Input* data tanggal produksi.
5. *Input* data tanggal selesai produksi.
6. *Input* data jumlah barang sebenarnya dari hasil produksi.
7. *Dropdown List* status produksi.
8. Tombol simpan data *Production Order*.

Home / Pp Production Order / 43 / Update

## Update Pp Production Order: 43

Production Request ID **1**

12

Material Produced **2**

Frame for Deluxe Touring Bike with Red Color

Quantity **3**

100

Production Date **4**

2016-06-25 00:00:00

Finish Date **5**

Choose Finish Date


Real Quantity **6**

Real Quantity

Status **7**

DLV

Update **8**



Gambar 4.11 Implementasi Antarmuka Formulir *Confirmation*

#### 4.5.6. Implementas Antarmuka Formulir *Input data Goods Receive*

Halaman antarmuka *input data Goods Receive* ditunjukkan oleh gambar 4.12. Berikut penjelasan bagian-bagian antarmuka *input data Goods Receive*:

1. *Dropdown List* yang berisi pilihan *Production Order ID*.
2. *Input data* tanggal *Goods Receive*.
3. *Dropdown List* dari nama-nama barang di dalam *Goods Receive*.
4. *Input data* jumlah barang yang direncanakan.
5. *Input data* jumlah barang yang sebenarnya.
6. Tombol hapus data barang.
7. Tombol tambah data barang.
8. Tombol simpan data *Goods Receive*.

Home / Pp Production Goods Receive / 2 / Update

## Update Pp Production Goods Receive: 2

Production Order ID **1**

2

Receive Date **2**

2016-06-25 00:00:00

Storage ID

4

**Pp Production Material** Showing 1-2 of 2 items.

#	Pp production request	lwm item master	Planned Quantity	Actual Quantity
1	1	Aluminium <b>3</b>	500 <b>4</b>	500 <b>5</b>
2	1	Paint drum with Red Color 20KG	5	5

**+Add Row** **7**

**Update** **8**

**6**

**Gambar 4.12 Implementasi Antarmuka Input Data *Goods Receive***

#### **4.5.7. Implementasi Antarmuka Formulir *Input data Goods Transfer***

Halaman antarmuka *input data Goods Transfer* ditunjukkan oleh gambar 4.13. Berikut penjelasan bagian-bagian antarmuka *input data Goods Transfer*:

1. *Dropdown List* yang berisi pilihan *Production Order ID*.
2. *Dropdown List* yang berisi barang hasil produksi.
3. *Input* data jumlah barang hasil produksi.
4. *Input* data tanggal *Goods Transfer*.
5. Tombol simpan data *Goods Transfer*.

Home / Pp Production Goods Transfer / Create Pp Production Goods Transfer

## Create Pp Production Goods Transfer

**Production Order ID** 1

**Material ID** 2

**Quantity** 3

**Storage ID**

**Transfer Date** 4

**Create** 5

**Gambar 4.13 Implementasi Antarmuka Input Data *Goods Transfer***

### 4.5.8. Implementasi Antarmuka *Generating Scheduling*

Implementasi halaman antarmuka *Generating Scheduling* ditunjukkan pada gambar 4.14. Berikut penjelasan bagian-bagian antarmuka halaman *Generating Scheduling* :

1. *Static Input* nama barang.
2. *Static Input* jumlah barang.
3. *Static Input* tanggal produksi.
4. *Static Input* tanggal selesai produksi.
5. *Button* pilihan *Convert*.

## Pp Production Request

Create Pp Production Request   Advance Search

■ Pp Production Request   Showing 1-3 of 3 items

Page   List   Full

#	Pp Material	Quantity	Production Date	Finish Date	Convert to P.O
		<input type="text"/>	<input type="text"/>	<input type="text"/>	
1	Professional Bike with Green Main Color <b>1</b>	10 <b>2</b>	2016-06-28 20:22:56 <b>3</b>	2016-06-28 20:22:56 <b>4</b>	Convert → <b>5</b>
2	Deluxe Touring Bike with Green Main Color	100	2016-06-23 06:13:00	2016-06-23 09:32:00	Converted →
3	Professional Bike with Red Color	100	2016-06-23 02:54:00	2016-06-23 06:08:00	Converted →

**Gambar 4.14 Implementasi Antarmuka Input Data *Goods Receive***

## 4.6. Implementasi Program Modul Produksi

Pada bagian ini akan dijelaskan mengenai implementasi yang terdapat pada modul Produksi. Implementasi modul Produksi dibagi menjadi beberapa bagian, yaitu:

1. *Routing*.
2. *Operasi*.
3. *Bill of Material*.
4. *Request for Production*.
5. *Scheduling*.
6. *Confirmation*.
7. *Goods Receive*.
8. *Goods Transfer*.

### 4.6.1. Program *Routing*

Implementasi *Routing* meliputi *method insert/update* data dan menambah operasi-operasi penyusun *Routing*. *Method* membuat data baru *Routing* disajikan pada kode sumber 4.19.

```
public function actionCreate()
{
    $model = new PpRouting();

    if ($model->loadAll(Yii::$app->request->post()) &&
        $model->saveAll()) {
```

```

        return $this->redirect(['view', 'id' =>
$model->id]);
    } else {
        return $this->render('create', [
            'model' => $model,
        ]);
    }
}

```

**Kode Sumber 4.19 Method Membuat Data Routings**

## 4.6.2. Program Operasi

Implementasi Operasi meliputi *method insert/update* data Operasi. *Method* membuat data baru Operasi disajikan pada kode sumber 4.20.

```

public function actionCreate()
{
    $model = new PpOperation();

    if ($model->loadAll(Yii::$app->request->post()) &&
$model->saveAll()) {
        return $this->redirect(['view', 'id' =>
$model->id]);
    } else {
        return $this->render('create', [
            'model' => $model,
        ]);
    }
}

```

**Kode Sumber 4.20 Method Membuat Data Operasi**

## 4.6.3. Program *Bill of Material*

Implementasi *Bill of Material* meliputi *method insert/update* data dan menambah material-material penyusun *Bill of Material*. *Method* membuat data baru *Bill of Material* disajikan pada kode sumber 4.31.

```

public function actionCreate()
{
    $model = new PpBom();

    if ($model->loadAll(Yii::$app->request->post()) &&
$model->saveAll()) {
        return $this->redirect(['view', 'id' =>

```

```

$model->id]);
    } else {
        return $this->render('create', [
            'model' => $model,
        ]);
    }
}

```

**Kode Sumber 4.21 Method Membuat Data *Bill of Material***

#### 4.6.4. Program *Request for Production*

Implementasi *Request for Production* meliputi *method insert/update data Request for Production*. *Method* membuat data baru *Request for Production* disajikan pada kode sumber 4.22.

```

public function actionCreate()
{
    $model = new PpProductionRequest();

    if ($model->loadAll(Yii::$app->request->post()) &&
        $model->saveAll()) {

        $production_material = new
        ProductionMaterial($model->material_produced, $model->id);
        $production_material->
        automateProductionMaterial();
        $production_material->
        insertProductionMaterial();

        return $this->redirect(['view', 'id' =>
        $model->id]);
    } else {
        return $this->render('create', [
            'model' => $model,
        ]);
    }
}

```

**Kode Sumber 4.22 Method Menambah Data *Request for Production***

#### 4.6.5. Program *Confirmation*

Program *Confirmation* bertujuan untuk melakukan depresiasi harga mesin yang dikelola oleh Modul *Asset Management* setiap suatu proses produksi telah selesai dilakukan. Implementasi program *Confirmation* ditunjukkan melalui Kode Sumber 4.23.

```

public function actionUpdate($id)
{
    $model = $this->findModel($id);

    $current_status = $model->status;

    if ($model->loadAll(Yii::$app->request->post()) &&
    $model->saveAll()) {

        $new_po = Yii::$app->request-
        >post('PpProductionOrder');

        if($current_status == "REL" &&
        $new_po['status'] == "CNF")

            ProductionRequest::subtractMachineLifeTime($id);

        return $this->redirect(['view', 'id' =>
        $model->id]);
    } else {
        return $this->render('update', [
            'model' => $model,
        ]);
    }
}

```

**Kode Sumber 4.23 Method Confirmation**

#### 4.6.6. Program *Goods Receive*

Program *goods receive* berfungsi untuk mencatat transaksi perpindahan material bahan produksi dari Modul *Inventory and Warehouse Management* ke Modul Produksi. Implementasi *insert data Goods Receive* ditunjukkan pada kodesumber 4.24.

```

public function actionUpdate($id)
{
    $model = $this->findModel($id);

    if ($model->loadAll(Yii::$app->request->post()) &&
    $model->saveAll()) {
        $data_line = PpProductionMaterial::find()-
        >where('production_order id =
        '.$model['production_order_id']->all();
        foreach ($data_line as $key => $value) {

```



```

IwmquantController::actionItemUse($value['production_order_id'], $value['material_id']);
    }
    return $this->redirect(['view', 'id' =>
$model->id]);
    } else {
        return $this->render('update', [
            'model' => $model,
        ]);
    }
}

```

**Kode Sumber 4.24 Method Membuat Goods Receive**

#### 4.6.7. Program Goods Transfer

Program *Goods Transfer* berfungsi untuk mencatat transaksi barang hasil produksi dari Modul Produksi ke Modul *Inventory and Warehous Management*. Implementasi *insert* data *Goods Transfer* ditunjukkan pada kodesumber 4.25.

```

public function actionCreate($id=null)
{
    $model = new PpProductionGoodsTransfer();

    if($id!=null){

        $order = PpProductionOrder::findOne($id);

        date_default_timezone_set("Asia/Jakarta");

        $model->production_order_id = $order->id;
        $model->material_id = $order->
material_produced;
        $model->quantity = $order->quantity;
        $model->storage_id = '2';
        $model->transfer_date = date("Y-m-d H:i:s");
    }

    if ($model->loadAll(Yii::$app->request->post()) &&
$model->saveAll()) {
        $type = 'Production';
        $production_order_id = $model->
production_order_id;
        $date = $model->transfer_date;
        $item = array(
            array(
                'item_id' => $model->

```

```

>material_id,
                                'item_qty' => $model->quantity
                                )
                                );
                                $create_iwm_receipt =
IwmgoodreceiptController::actionCreateReceipt($type,$production_order_id,$date,$item);

                                $production_order =
PpProductionOrder::findOne($model->production_order_id);
                                $production_order->status = 'DLV';
                                $production_order->save();

                                return $this->redirect(['view', 'id' =>
$model->id]);
                                } else {
                                return $this->render('create', [
                                'model' => $model,
                                ]);
                                }
                                }

```

**Kode Sumber 4.25 Method Membuat Goods Transfer**

## 4.7. Implementasi Program *Scheduling*

### 4.7.1. Pengkodean Kromosom

Implementasi pengkodean kromosom dilakukan pada kelas *Scheduling* dan *OptimizedStringGenerator*. Pemilihan operasi-operasi yang terlibat dilakukan pada *method \_\_construct()* pada kelas *Scheduling*. Data-data operasi diambil dari tabel *pp\_operation*, *pp\_routing\_wc\_op* dan *pp\_operation* di basis data berdasarkan barang yang akan diporduksi. Sedangkan, penyusunan kromosom dilakukan pada kelas *OptimizedStringGenerator*. Pada kelas ini terdapat tahap-tahap *non-delay permutation encoding* yang dirangkum di dalam *method generateHybrid()*. Pengkodean kromosom disajikan pada kode sumber 4.26.

```

class OptimizedStringGenerator
{
    public function generateHybrid(){
        $maxGen = 20;

```

```

        $population = array();
        while($maxGen--){
            $string = array();
            $stable = $this->createString();
            //Langkah 1
            $frontJobs = self::getFront($stable);

            while(!empty($frontJobs)){

                /*Langkah 2.1*/
                $fastest =
self::getFastest($frontJobs);
                /*Langkah 2.2*/
                $commRes =
self::getCommon($frontJobs,$frontJobs[$fastest]);
                /*Langkah 2.3*/
                $earliest =
self::getEarliest($commRes);
                /*Langkah 2.4*/
                $finalB =
self::deleteOnB($commRes,$commRes[$earliest]);
                /*Langkah 2.5*/
                $frontJobs =
self::deleteFromA($frontJobs,$finalB[0]);
                /*Langkah 3*/
                $selectedOp =
self::updateTime($string,$finalB[0]);
                array_push($string,$selectedOp);
                /*Langkah 4*/
                $frontJobs =
self::addToA($frontJobs,$stable[$selectedOp['nth_job']], $se
lectedOp);
                array_push($population,array("string"=>$string));
            }return $population;}
    }

```

**Kode Sumber 4.26 Pengkodean Kromosom**

#### 4.7.2. Pindah Silang

Implementasi Pindah Silang dilakukan pada kelas *GA*. Pindah silang diimplementasikan pada *method crossOver* dengan parameter berupa 2 buah kromosom orangtua dan memiliki *return* berupa hasil persilangan. Sedangkan, implementasi *Roulette Wheele Selection* dilakukan pada *method rouletteWheel*. Implementasi pindah silang dan *Roulette Wheele Selection* digambarkan pada kode sumber 4.27.

```

function crossOver($p1 = array(), $p2 = array()){
    if(mt_rand(1,100) <= $this->crossOverProb){
        while(isset($parent1[$index1]) and
isset($parent2[$index2])){$rand_parent = mt_rand(1,2);
            if($rand_parent == 1){
                while(isset($parent1[$index1]) and
in_array($parent1[$index1]['nth_job'], $pool)){$parent1[$index1]
['nth_operation'], $pool)}{$index1++;}

                if(isset($parent1[$index1])){
                    $offspring[]
= $parent1[$index1];
                    $pool[] =
$parent1[$index1]['nth_job'], $parent1[$index1]['nth_op
eration'];}}
                    else if($rand_parent == 2){
                        while(isset($parent2[$index2]) and
in_array($parent2[$index2]['nth_job'], $pool)){$parent2[$index2]
['nth_operation'], $pool)}{$index2++;}
                        if(isset($parent2[$index2])){
                            $offspring[]
= $parent2[$index2]; $pool[] =
$parent2[$index2]['nth_job'], $parent2[$index2]['nth_op
eration'];
                        }}}}return
$offspring;}else{
    if($p1["fitness"]>$p2["fitness"])
        return $p1["string"];
    else return $p2["string"];
}}
function rouletteWheel(){
    $this->total_scalabled_fitness = 0;
    $start_r = 0;
    foreach($this->population as $key =>
$individual){
        $scalabled =
round($individual['fitness']*10000/$this->total_fitness);
        $this->
population[$key]['roulette_range']['start'] = $start_r;
        $end_r = $start_r + $scalabled;
        $start_r = $end_r;
        $this->
population[$key]['roulette_range']['end'] = $end_r;
        $this->total_scalabled_fitness +=
$scalabled;
    }}

```

**Kode Sumber 4.27 Pindah Silang dan *Roulette Wheel***

### 4.7.3. Mutasi

Implementasi mutasi dilakukan pada kelas *GA*. Mutasi diimplementasikan pada *method mutation()* yang memiliki parameter berupa kromosom hasil pindah silang dan memiliki *return* berupa kromosom hasil mutasi. Implementasi mutasi digambarkan pada kode sumber 4.28.

```
function mutation($string){
    if(mt_rand(1,100) <= $this->mutationProb){
        $length = count($string)-1;
        $gen1 = mt_rand(0,$length);
        $gen2 = mt_rand(0,$length);
        $temp = $string[$gen1];
        $string[$gen1] = $string[$gen2];
        $string[$gen2] = $temp;
    }
    return $string;
}
```

**Kode Sumber 4.28 Implementasi Mutasi**

### 4.7.4. Fitness Function

Implementasi *fitness function* dilakukan pada kelas *GA* di dalam *method getFitness()*. Implementasi *fitness function* digambarkan pada kode sumber 4.29.

```
function getFitness(){
    $this->total_fitness = 0;
    foreach($this->population as $key =>
$individual){
        $fitness =
self::getFitness2($individual["string"], $key);
        $this->population[$key]['fitness'] =
$fitness;
        $this->total_fitness += $fitness;
    }
    //self::printSomething($this-
>total_fitness);
}

function getFitness2($string,$individual_key){
    $start = microtime(true);
    foreach($string as $key => $gen){
        $table[$gen['nth_job']][] = $gen;
```

```
    }
    $fitness = 0;
    foreach($table as $row){
        $first_job = $row[0];
        $last_job = $row[count($row)-1];
        $fitness +=
        (($last_job['process_time']+$last_job['starting_time'])-
        $first_job['starting_time']);
    }
    $fitness = 1 / ($fitness / count($table));

    return $fitness;
}
```

**Kode Sumber 4.29 Implementasi *Fitness Function***

## BAB V

### PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada aplikasi yang dikembangkan. Pengujian yang dilakukan adalah pengujian terhadap kebutuhan fungsionalitas sistem yang telah dijabarkan pada Bab III dan terhadap tujuan dibuatnya aplikasi ini, yakni manajemen produksi pada ERP.

#### 5.1. Lingkungan Pengujian

Lingkungan pengujian sistem pada pengerjaan Tugas Akhir ini dilakukan pada lingkungan dan alat kakas sebagai berikut:

Prosesor : Intel® Core™ i3-CPU (2.20 GHz)  
 RAM : 2 GB  
 Jenis *Device* : *Notebook*  
 Sistem Operasi : Microsoft Windows 7 Ultimate

#### 5.2. Skenario Pengujian Fungsionalitas Modul Produksi

##### 5.2.1. Pengujian Fitur Mengelola data *Routing*

Pada subbab ini akan dijelaskan mengenai hasil pengujian fitur mengelola data *Routing* yang meliputi skenario penambahan, perubahan, dan penghapusan data.

<b>ID</b>	UJ-UC-001
<b>Referensi Kasus Penggunaan</b>	UC-001
<b>Nama</b>	Pengujian Fitur Mengelola <i>Routing</i>
<b>Tujuan Pengujian</b>	Menguji kemampuan aplikasi mengelola data <i>Routing</i>
<b>Skenario 1</b>	<i>Pengguna menambahkan data Routing</i>
<b>Kondisi Awal</b>	Aktor berada pada halaman membuat data <i>Routing</i>

<b>Data Uji</b>	Inputan data <i>Routing</i> berupa <i>Material Produced</i> : Tire for All Bike, <i>List of Operation</i> : 1. <i>Wheel Assembly (Material Staging)</i> , 2. <i>Wheel Assembly : (Assemble Components)</i> , 3. <i>Wheel Assembly : (Move to Storage)</i>
<b>Langkah Pengujian</b>	Aktor memasukkan Data Uji ke dalam <i>form</i> yang tersedia dan memilih pilihan simpan data.
<b>Hasil Yang Diharapkan</b>	Data yang dimasukkan ke dalam basis data berhasil dimasukkan.
<b>Hasil Yang Didapat</b>	Data yang dimasukkan tersimpan di basis data
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Aktor berada pada halaman melihat data yang baru saja dimasukkan.
<b>Skenario 2</b>	<b><i>Aktor menyunting data Routing</i></b>
<b>Kondisi Awal</b>	Aktor berada pada halaman utama daftar <i>Routing</i>
<b>Data Uji</b>	Perubahan data <i>Routing Tire for All Bike</i> dengan menghapus operasi terakhir, yaitu <i>Wheel Assembly : (Move to Storage)</i> .
<b>Langkah Pengujian</b>	Aktor memilih menyunting salah satu data <i>Routing</i> yang ada pada daftar, lalu menyunting data tersebut pada halaman menyunting dan menyimpannya kembali ke basis data.
<b>Hasil Yang Diharapkan</b>	Data yang disunting akan berubah dan disimpan kembali ke basis data.
<b>Hasil Yang Didapat</b>	Data yang disunting tersimpan di basis data
<b>Hasil Pengujian</b>	Berhasil.
<b>Kondisi Akhir</b>	Aktor berada pada halaman melihat data yang baru saja disunting.
<b>Skenario 3</b>	<b><i>Aktor menghapus data Routing</i></b>
<b>Kondisi Awal</b>	Aktor berada pada halaman utama <i>Routing</i>
<b>Data Uji</b>	Data <i>Routing Tire for All Bike</i>
<b>Langkah Pengujian</b>	Pengguna memilih menghapus salah satu data <i>Routing</i> yang ada pada daftar, lalu melakukan konfirmasi penghapusan data tersebut dan data akan dihapus oleh sistem.
<b>Hasil Yang Diharapkan</b>	Data yang dihapus akan terhapus di basis data



<b>Hasil Yang Didapat</b>	Data berhasil di hapus
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Pengguna berada pada halaman utama <i>Routing</i>
<b>Skenario 4</b>	<b><i>Pengguna melihat data Routing</i></b>
<b>Kondisi Awal</b>	Aktor berada pada halaman utama <i>Routing</i>
<b>Data Uji</b>	Data <i>Routing Frame for Professional Bike with Red Color</i>
<b>Langkah Pengujian</b>	Aktor masuk ke halaman daftar <i>Routing</i> dan memilih salah satu data untuk ditampilkan.
<b>Hasil Yang Diharapkan</b>	Data yang dimasukkan ke dalam basis data berhasil ditampilkan.
<b>Hasil Yang Didapat</b>	Data <i>Routing</i> ditampilkan
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Aktor berada pada halaman melihat data <i>Routing</i> .

Pengujian pada skenario 1 berupa penambahan data *Routing* berhasil dijalankan seperti yang digambarkan pada gambar 5.1. Skenario 2 berupa penyuntingan data *Routing* berhasil dijalankan seperti yang digambarkan pada gambar 5.2. Skenario 3 berupa penghapusan data *Routing* berhasil dijalankan seperti yang digambarkan pada gambar 5.3 serta gambar 5.4. Skenario 4 melihat data *Routing* berhasil dijalankan seperti yang digambarkan pada gambar 5.5.

Pp Routing 22 PDF Update Delete

ID	22
Iwm Item Master	Tire for All Bike

Pp Routing Wc Op 22 Showing 1-3 of 3 items.

#	Pp Routing	Pp Operation
1	22	Wheel Assembly : Material Staging
2	22	Wheel Assembly : Assemble Components
3	22	Wheel Assembly : Move to Storage

**Gambar 5.1 Pengujian Penambahan Data *Routings Tire for Al Bike***

Pp Routing 22 PDF Update Delete











ID	22
Iwm Item Master	Tire for All Bike

Pp Routing Wc Op 22 Showing 1-2 of 2 items.

#	Pp Routing	Pp Operation
1	22	Wheel Assembly : Material Staging
2	22	Wheel Assembly : Assemble Components

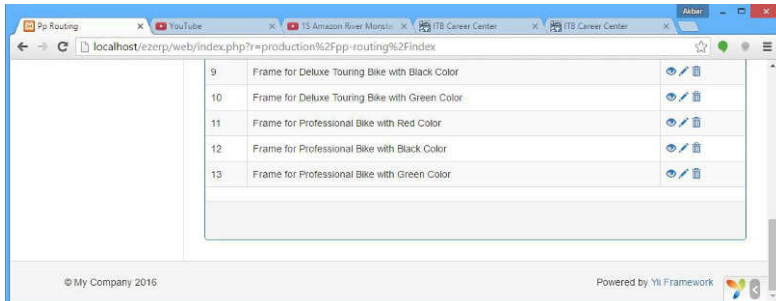
**Gambar 5.2 Penyuntingan Data *Routings Tire for All Bike***

Pp Routing

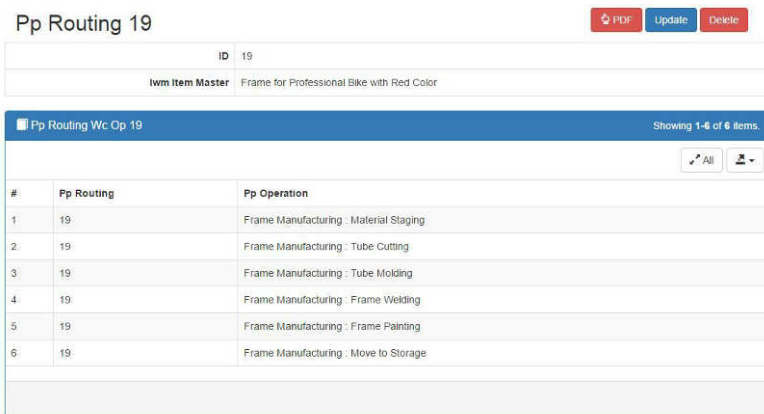
10	Frame for Deluxe Touring Bike with Green Color	 
11	Frame for Professional Bike with Red Color	 
12	Frame for Professional Bike with Black Color	 
13	Frame for Professional Bike with Green Color	 
14	Tire for All Bike	 

© My Company 2016 Powered by Yii Framework

**Gambar 5.3 Data Sebelum Penghapusan *Routings Tire for All Bike***



**Gambar 5.4 Setelah Penghapusan Data *Routings Tire for All Bike***



**Gambar 5.5 Melihat Data *Routings***

### 5.2.2. Pengujian Fitur Mengelola Data Operasi

Pada subbab ini akan dijelaskan mengenai hasil pengujian fitur mengelola data Operasi yang meliputi skenario penambahan, perubahan, dan penghapusan data.

<b>ID</b>	UJ-UC-002
<b>Referensi Kasus Penggunaan</b>	UC-002

<b>Nama</b>	Pengujian Fitur Mengelola Operasi
<b>Tujuan Pengujian</b>	Menguji kemampuan aplikasi mengelola data Operasi
<b>Skenario 1</b>	<b><i>Pengguna menambahkan data Operasi</i></b>
<b>Kondisi Awal</b>	Aktor berada pada halaman membuat data Operasi
<b>Data Uji</b>	<i>Input data Operasi baru Checking Material dengan type : Human Based</i>
<b>Langkah Pengujian</b>	Aktor memasukkan Data Uji ke dalam <i>form</i> yang tersedia dan memilih pilihan simpan data.
<b>Hasil Yang Diharapkan</b>	Data yang dimasukkan ke dalam basis data berhasil dimasukkan.
<b>Hasil Yang Didapat</b>	Data yang dimasukkan tersimpan di basis data
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Aktor berada pada halaman melihat data yang baru saja dimasukkan.
<b>Skenario 2</b>	<b><i>Aktor menyunting data Operasi</i></b>
<b>Kondisi Awal</b>	Aktor berada pada halaman utama daftar Operasi
<b>Data Uji</b>	Data Operasi <i>Checking Material</i>
<b>Langkah Pengujian</b>	Aktor memilih menyunting salah satu data Operasi yang ada pada daftar, lalu menyunting data tersebut pada halaman menyunting dan menyimpannya kembali ke basis data.
<b>Hasil Yang Diharapkan</b>	Data yang disunting akan berubah dan disimpan kembali ke basis data.
<b>Hasil Yang Didapat</b>	Data yang disunting tersimpan di basis data
<b>Hasil Pengujian</b>	Berhasil.
<b>Kondisi Akhir</b>	Aktor berada pada halaman melihat data yang baru saja disunting.
<b>Skenario 3</b>	<b><i>Aktor menghapus data Operasi</i></b>
<b>Kondisi Awal</b>	Aktor berada pada halaman utama Operasi
<b>Data Uji</b>	Data Operasi <i>Checking Material 1</i>
<b>Langkah Pengujian</b>	Pengguna memilih menghapus salah satu data Operasi yang ada pada daftar, lalu melakukan konfirmasi penghapusan data tersebut dan data akan dihapus oleh

	sistem.
Hasil Yang Diharapkan	Data yang dihapus akan terhapus di basis data
Hasil Yang Didapat	Data berhasil di hapus
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman utama Operasi
Skenario 4	<i>Pengguna melihat data Operasi</i>
Kondisi Awal	Aktor berada pada halaman utama Operasi
Data Uji	Data Operasi <i>Wheel Assembly : Material Staging</i>
Langkah Pengujian	Aktor masuk ke halaman daftar Operasi dan memilih salah satu data untuk ditampilkan.
Hasil Yang Diharapkan	Data yang dimasukkan ke dalam basis data berhasil ditampilkan.
Hasil Yang Didapat	Data Operasi ditampilkan
Hasil Pengujian	Berhasil
Kondisi Akhir	Aktor berada pada halaman melihat data Operasi.

Pengujian pada skenario 1 berupa penambahan data Operasi berhasil dijalankan seperti yang digambarkan pada gambar 5.6. Skenario 2 berupa penyuntingan data Operasi berhasil dijalankan seperti yang digambarkan pada gambar 5.7 dan 5.8. Skenario 3 berupa penghapusan data Operasi berhasil dijalankan seperti yang digambarkan pada gambar 5.9 dan gambar 5.10. Skenario 4 melihat data Operasi berhasil dijalankan seperti yang digambarkan pada gambar 5.11.



Gambar 5.6 Penambahan Data Operasi

## Pp Operation 22

[PDF](#) [Update](#) [Delete](#)

ID	22
Operation	Checking Material
Type	Human Based

**Gambar 5.7 Sebelum Perubahan Data Operasi *Checking Material***

## Pp Operation 22

[PDF](#) [Update](#) [Delete](#)

ID	22
Operation	Checking Material 1
Type	Machine Based

**Gambar 5.8 Setelah Perubahan Data Operasi *Checking Material***

## Pp Operation

[Create Pp Operation](#)
[Advance Search](#)

Pp Operation			Showing 21-21 of 21 Items
#	Operation	Type	
21	Checking Material 1	Machine Based	<a href="#">View</a> <a href="#">Edit</a>

**Gambar 5.9 Sebelum Penghapusan Data Operasi ID 21**

16	Frame Manufacturing : Tube Cutting	Collaboration	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
17	Frame Manufacturing : Tube Molding	Collaboration	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
18	Frame Manufacturing : Frame Welding	Collaboration	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
19	Frame Manufacturing : Frame Painting	Collaboration	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
20	Frame Manufacturing : Move to Storage	Human Based	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>

**Gambar 5.10 Setelah Penghapusan Data Operasi ID 21**

Pp Operation 1		PDF	Update	Delete
ID	1			
Operation	Wheel Assembly - Material Staging			
Type	Human Based			

**Gambar 5.11 Melihat Data Operasi ID 1**

### 5.2.3. Pengujian Fitur Mengelola data *Bill of Material*

Pada subbab ini akan dijelaskan mengenai hasil pengujian fitur mengelola data *Bill of Material* yang meliputi skenario penambahan, perubahan, dan penghapusan data.

<b>ID</b>	UJ-UC-003
<b>Referensi Kasus Penggunaan</b>	UC-003
<b>Nama</b>	Pengujian Fitur Mengelola <i>Bill of Material</i>
<b>Tujuan Pengujian</b>	Menguji kemampuan aplikasi mengelola data <i>Bill of Material</i>
<b>Skenario 1</b>	<b><i>Pengguna menambahkan data Bill of Material</i></b>
<b>Kondisi Awal</b>	Aktor berada pada halaman membuat data <i>Bill of Material</i>
<b>Data Uji</b>	Input data baru <i>Bill of Material</i> Material ID : Tire for All Bike, description : Tire for All Kind of Bike, Daftar Material : 1). Item : Hex Nut 5mm/UOM:(Pcs)Pieces, Quantity: 1 dan 2). Item : Socket Head Bolt 5mm/UOM:(Pcs)Pieces, Quantity: 1
<b>Langkah Pengujian</b>	Aktor memasukkan Data Uji ke dalam form yang tersedia dan memilih pilihan simpan data.
<b>Hasil Yang Diharapkan</b>	Data yang dimasukkan ke dalam basis data berhasil dimasukkan.
<b>Hasil Yang Didapat</b>	Data yang dimasukkan tersimpan di basis data
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Aktor berada pada halaman melihat data yang baru saja dimasukkan.
<b>Skenario 2</b>	<b><i>Aktor menyunting data Bill of Material</i></b>
<b>Kondisi Awal</b>	Aktor berada pada halaman utama daftar <i>Bill of Material</i> .

<b>Data Uji</b>	Data <i>Bill of Material</i> : <i>Tire for All Bike</i> dengan perubahan data kolom <i>description</i> menjadi <i>Tire for Professional Bike</i> .
<b>Langkah Pengujian</b>	Aktor memilih menyunting salah satu data <i>Bill of Material</i> yang ada pada daftar, lalu menyunting data tersebut pada halaman menyunting dan menyimpannya kembali ke basis data.
<b>Hasil Yang Diharapkan</b>	Data yang disunting akan berubah dan disimpan kembali ke basis data.
<b>Hasil Yang Didapat</b>	Data yang disunting tersimpan di basis data.
<b>Hasil Pengujian</b>	Berhasil.
<b>Kondisi Akhir</b>	Aktor berada pada halaman melihat data yang baru saja disunting.
<b>Skenario 3</b>	<b><i>Aktor menghapus data Bill of Material</i></b>
<b>Kondisi Awal</b>	Aktor berada pada halaman utama <i>Bill of Material</i> .
<b>Data Uji</b>	Data <i>Bill of Material</i> : <i>Tire for All Bike</i> .
<b>Langkah Pengujian</b>	Pengguna memilih menghapus salah satu data <i>Bill of Material</i> yang ada pada daftar, lalu melakukan konfirmasi penghapusan data tersebut dan data akan dihapus oleh sistem.
<b>Hasil Yang Diharapkan</b>	Data yang dihapus akan terhapus di basis data.
<b>Hasil Yang Didapat</b>	Data berhasil di hapus.
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Pengguna berada pada halaman utama <i>Bill of Material</i> .
<b>Skenario 4</b>	<b><i>Pengguna melihat data Bill of Material</i></b>
<b>Kondisi Awal</b>	Aktor berada pada halaman utama <i>Bill of Material</i> .
<b>Data Uji</b>	Data <i>Bill of Material</i> : <i>Deluxe Touring Bike with Black Main Color</i>
<b>Langkah Pengujian</b>	Aktor masuk ke halaman daftar <i>Bill of Material</i> dan memilih salah satu data untuk ditampilkan.
<b>Hasil Yang Diharapkan</b>	Data yang dimasukkan ke dalam basis data berhasil ditampilkan.
<b>Hasil Yang Didapat</b>	Data <i>Bill of Material</i> ditampilkan.



<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Aktor berada pada halaman melihat data <i>Bill of Material</i> .

Pengujian pada skenario 1 berupa penambahan data *Bill of Material* berhasil dijalankan seperti yang digambarkan pada lampiran 5.12. Skenario 2 berupa penyuntingan data *Bill of Material* berhasil dijalankan seperti yang digambarkan pada gambar 5.13. Skenario 3 berupa penghapusan data *Bill of Material* berhasil dijalankan seperti yang digambarkan pada gambar 5.14 dan gambar 5.15. Skenario 4 melihat data *Bill of Material* berhasil dijalankan seperti yang digambarkan pada gambar 5.16.

Pp Bom 22 PDF Update Delete

ID	22
Iwm Item Master	Tire for All Bike
Description	Tire for Professional Bike

**Pp Bom Material 22** Showing 1-2 of 2 items.

#	Pp Bom	Iwm Item Master	Quantity
1	22	Hex Nut 5mm	1
2	22	Socket Head Bolt 5mm	1

**Gambar 5.12 Penambahan Data *BOM Tire for All Bike***

Pp Bom 22 PDF Update Delete

ID	22
Iwm Item Master	Tire for All Bike
Description	Tire for All Kind of Bike

**Pp Bom Material 22** Showing 1-2 of 2 items.

#	Pp Bom	Iwm Item Master	Quantity
1	22	Hex Nut 5mm	1
2	22	Socket Head Bolt 5mm	1

**Gambar 5.13 Setelah Perubahan Data *BOM Tire for All Bike***

12	Professional Bike with Black Color	PRFS - Black - Professional Bike - Primary BOM	
13	Professional Bike with Green Main Color	PRFS - Green - Professional Bike - Primary BOM	
14	Tire for All Bike	Tire for Professional Bike	

© My Company 2016 Powered by Yli Framework

**Gambar 5.14 Sebelum Penghapusan Data *BOM Tire for All Bike***

11	Professional Bike with Red Color	PRFS - Red - Professional Bike - Primary BOM	
12	Professional Bike with Black Color	PRFS - Black - Professional Bike - Primary BOM	
13	Professional Bike with Green Main Color	PRFS - Green - Professional Bike - Primary BOM	

© My Company 2016 Powered by Yli Framework

**Gambar 5.15 Setelah Penghapusan Data *BOM Tire for All Bike***

Pp Bom 8		PDF	Update	Delete
ID	8			
Iwm Item Master	Deluxe Touring Bike with Black Main Color			
Description	DXTR - Black - Deluxe Touring Bike - Primary BOM			

Pp Bom Material 8			Showing 1-8 of 8 items.
			<input type="checkbox"/> All <input type="checkbox"/>
#	Pp Bom	Iwm Item Master	Quantity
1	8	Frame for Deluxe Touring Bike with Black Color	1
2	8	Wheel's Bike	2
3	8	Chain's Bike	1
4	8	Handle Bar's Bike	1
5	8	Seat Kit's Bike	1
6	8	Pedal's Bike	2
7	8	Gear's Bike	1
8	8	Front and Rear Bike's Bike	1

Gambar 5.16 Melihat Data BOM ID 8

#### 5.2.4. Pengujian Fitur Membuat *Request for Production*

Pada subbab ini akan dijelaskan mengenai hasil pengujian fitur menambahkan data *Request for Production* yang meliputi skenario penambahan data.

<b>ID</b>	UJ-UC-004
<b>Referensi Kasus Penggunaan</b>	UC-004
<b>Nama</b>	Pengujian Fitur Mengelola <i>Request for Production</i>
<b>Tujuan Pengujian</b>	Menguji kemampuan aplikasi mengelola data <i>Request for Production</i>
<b>Skenario 1</b>	<b>Pengguna menambahkan data <i>Request for Production</i></b>
<b>Kondisi Awal</b>	Aktor berada pada halaman membuat data <i>Request for Production</i>
<b>Data Uji</b>	Data <i>Request for Production</i> baru dengan isian kolom sebagai berikut : 1). <i>Material Produced</i> : <i>Deluxe Touring Bike with Red Main Color</i> , 2). <i>Quantity</i> : 100 3). <i>Production Date</i> : 2016-06-27 00:00:00, 3). <i>Finish Date</i> : 2016-06-27 00:00:00.

<b>Langkah Pengujian</b>	Aktor memasukkan Data Uji ke dalam <i>form</i> yang tersedia dan memilih pilihan simpan data.
<b>Hasil Yang Diharapkan</b>	Data yang dimasukkan ke dalam basis data berhasil dimasukkan.
<b>Hasil Yang Didapat</b>	Data yang dimasukkan tersimpan di basis data
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Aktor berada pada halaman melihat data yang baru saja dimasukkan.

Pengujian pada skenario 1 berupa penambahan data *Request for Production* berhasil dijalankan seperti yang digambarkan pada gambar 5.17.

Pp Production Request 34

[PDF](#)
[Update](#)
[Delete](#)

ID	34
Iwm Item Master	Deluxe Touring Bike with Red Main Color
Quantity	100
Production Date	2016-06-27 00:00:00
Finish Date	2016-06-27 00:00:00

**Gambar 5.17 Penambahan Data *Request for Production : Deluxe Touring Bike with Red Main Color***

### 5.2.5. Pengujian Fitur *Confirmation*

Pada subbab ini akan dijelaskan mengenai hasil pengujian fitur *Confirmation* yang meliputi skenario melakukan *confirmation*.

<b>ID</b>	UJ-UC-005
<b>Referensi Kasus Penggunaan</b>	UC-005
<b>Nama</b>	Pengujian Fitur <i>Confirmation</i>
<b>Tujuan Pengujian</b>	Menguji kemampuan aplikasi melakukan <i>confirmation</i>
<b>Skenario 1</b>	<i>Pengguna melakukan confirmation</i>
<b>Kondisi Awal</b>	Aktor berada pada halaman utama <i>Production Order</i>

<b>Data Uji</b>	Data <i>production order</i> ID 3, 100 buah <i>Professional Bike with Black Color</i> .
<b>Langkah Pengujian</b>	Aktor masuk ke halaman daftar data <i>Production Order</i> , memilih salah satu data uji, dan memilih pilihan <i>confirmation</i> pada data uji tersebut.
<b>Hasil Yang Diharapkan</b>	Data <i>usage time</i> pada setiap mesin yang terlibat akan bertambah.
<b>Hasil Yang Didapat</b>	Data <i>usage time</i> pada setiap mesin yang terlibat berhasil bertambah.
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Aktor berada pada halaman melihat data yang baru saja dilakukan <i>confirmation</i> .

Pengujian fitur *confirmation* melalui 1 skenario, yaitu melakukan *confirmation* dari sebuah data *Production Order*. Data uji yang digunakan adalah *production order* pembuatan 100 buah *Professional Bike with Black Color* sesuai pada gambar 5.18. Pembuatan produk tersebut akan melibatkan mesin-mesin pada tabel 5.1 dan gambar 5.19.

Pp Production Order 3

PDF
Update
Delete

ID	3
Pp Production Request	18
Iwm Item Master	Professional Bike with Black Color
Quantity	100
Pp Plant	(not set)
Storage ID	(not set)
Hrm Employee	(not set)
Production Date	2016-07-10 00:00:00
Finish Date	2016-07-10 02:52:00
Real Quantity	(not set)
Status	CRTD

**Gambar 5.18 Data *Confirmation* 100 buah *Professional Bike with Black Color***

#	Pp Operation	Am Asset Active	Hm Employee	Process Time in Minutes	
				<input type="text"/>	
1	Test bike	Testing Machine 1	E107	2	 
2	Test bike	Testing Machine 2	E108	3	 
3	Frame Manufacturing : Frame Welding	Welding Machine 1	E113	2	 
4	Frame Manufacturing : Frame Welding	Welding Machine 2	E114	2	 
5	Frame Manufacturing : Frame Painting	Spray Painting Machine 1	E115	2	 
6	Frame Manufacturing : Frame Painting	Spray Painting Machine 2	E116	3	 

**Gambar 5.19 Daftar Mesin yang Terlibat Pada Proses Produksi**

**Tabel 5.1 Daftar Mesin yang Terlibat**

No.	Operasi	Nama Mesin	Waktu Proses
1.	<i>Test Bike</i>	<i>Testing Machine 1</i>	2 menit
2.	<i>Test Bike</i>	<i>Testing Machine 2</i>	3 menit
3.	<i>Frame Welding</i>	<i>Welding Machine 1</i>	2 menit
4.	<i>Frame Welding</i>	<i>Welding Machine 2</i>	2 menit
5.	<i>Frame Painting</i>	<i>Spray Machine 1</i>	2 menit
6.	<i>Frame Painting</i>	<i>Spray Machine 2</i>	3 menit

Alokasi penggunaan mesin-mesin tersebut dilakukan secara otomatis oleh Algoritma Genetika melalui penjadwalan. Hasil penjadwalan yang didapatkan melalui *query SQL* pada *production order* ID 3 dijelaskan pada tabel 5.2.

**Tabel 5.2 Mesin yang Telah Digunakan Berdasar *Query***

No.	ID	Operasi	Nama Mesin	Waktu Digunakan
1.	1	<i>Test Bike</i>	<i>Testing Machine 1</i>	110 menit
2.	4	<i>Test Bike</i>	<i>Testing Machine 2</i>	135 menit
3.	8	<i>Frame Welding</i>	<i>Welding Machine 1</i>	94 menit
4.	3	<i>Frame Welding</i>	<i>Welding Machine 2</i>	106 menit
5.	10	<i>Frame Painting</i>	<i>Spray Machine 1</i>	108 menit
6.	7	<i>Frame Painting</i>	<i>Spray Machine 2</i>	92 menit

Pengujian dilakukan dengan membandingkan nilai *usage* pada setiap mesin disaat sebelum dan sesudah *confirmation*. Nilai

*usage* bertambah setelah melakukan *confirmation* pada setiap mesin sesuai pada tabel 5.3, gambar 5.20 dan gambar 5.21.

**Tabel 5.3 *Usage* Mesin Disaat Sebelum dan Sesudah *Confirmation***

<b>N o.</b>	<b>Nama Mesin</b>	<b><i>Usage</i> Sebelum</b>	<b><i>Usage</i> Sesudah</b>	<b>Kenaikan</b>
1.	<i>Testing Machine 1</i>	7:10:00	9:00:00	110 menit
2.	<i>Testing Machine 2</i>	6:30:00	8:45:00	135 menit
3.	<i>Welding Machine 1</i>	5:40:00	7:14:00	94 menit
4.	<i>Welding Machine 2</i>	4:56:00	6:42:00	106 menit
5.	<i>Spray Machine 1</i>	4:56:00	6:44:00	108 menit
6.	<i>Spray Machine 2</i>	5:53:00	7:25:00	92 menit

Register	Max Usage	Usage	Distance to Maintenance	Time Per Production	Capacity	Distance to Maintenance	Uom	Asset Description
	86400:00:00	7:10:00	0	00:10:00	6	0	(not set)	Testing Machine for Line 1
	86400:00:00	00:00:00	0	00:05:00	10	0	(not set)	Laser for cutting
	86400:00:00	4:56:00	0	00:07:00	8	0	(not set)	Welding Machine for Line 2
	86400:00:00	6:30:00	0	00:10:00	6	0	(not set)	Testing Machine for Line 2
	86400:00:00	00:00:00	0	00:05:00	10	0	(not set)	Laser for cutting
	86400:00:00	00:00:00	0	00:07:00	8	0	(not set)	Molding Machine for Line 1
	86400:00:00	5:53:00	0	00:07:00	8	0	(not set)	Machine used for frame painting 2
	86400:00:00	5:40:00	0	00:07:00	8	0	(not set)	Welding Machine for Line 1
	86400:00:00	00:00:00	0	00:07:00	8	0	(not set)	Molding Machine for Line 1
	86400:00:00	4:56:00	0	00:07:00	8	0	(not set)	Machine used for frame painting 1

**Gambar 5.20 Usage Setiap Mesin Sebelum Confirmation**



Register	Max Usage	Usage	Distance to Maintenance	Time Per Production	Capacity	Distance to Maintenance	Uom	Asset Description
	86400:00:00	9:00:00	0	00:10:00	6	0	(not set)	Testing Machine for Line 1
	86400:00:00	00:00:00	0	00:06:00	10	0	(not set)	Laser for cutting
	86400:00:00	6:42:00	0	00:07:00	8	0	(not set)	Welding Machine for Line 2
	86400:00:00	8:45:00	0	00:10:00	6	0	(not set)	Testing Machine for Line 2
	86400:00:00	00:00:00	0	00:06:00	10	0	(not set)	Laser for cutting
	86400:00:00	00:00:00	0	00:07:00	8	0	(not set)	Molding Machine for Line 1
	86400:00:00	7:25:00	0	00:07:00	8	0	(not set)	Machine used for frame painting 2
	86400:00:00	7:14:00	0	00:07:00	8	0	(not set)	Welding Machine for Line 1
	86400:00:00	00:00:00	0	00:07:00	8	0	(not set)	Molding Machine for Line 1
	86400:00:00	6:44:00	0	00:07:00	8	0	(not set)	Machine used for frame painting 1

Gambar 5.21 Usage Setiap Mesin Sesudah Confirmation

5.2.6. Pengujian Fitur Menambah Goods Receive

Pada subbab ini akan dijelaskan mengenai hasil pengujian fitur mengelola data Goods Receive yang meliputi skenario menambah data.

ID	UJ-UC-006
Referensi Kasus Penggunaan	UC-006
Nama	Pengujian Fitur Menambahkan Goods Receive

<b>Tujuan Pengujian</b>	Menguji kemampuan aplikasimenambah <i>Goods Receive</i>
<b>Skenario 1</b>	<b><i>Pengguna menambah Goods Receive</i></b>
<b>Kondisi Awal</b>	Aktor berada pada halaman utama <i>Production Order</i>
<b>Data Uji</b>	Data <i>Production Order</i> 82
<b>Langkah Pengujian</b>	Aktor masuk ke halaman daftar data <i>Production Order</i> , memilih salah satu data uji, memilih pilihan <i>goods receive</i> pada data uji tersebut, memasukkan data pada form, dan menyimpan data <i>goods receive</i> .
<b>Hasil Yang Diharapkan</b>	Data <i>goods receive</i> yang baru berhasil tersimpan dan data <i>goods issue</i> pada modul pergudangan juga berhasil tersimpan.
<b>Hasil Yang Didapat</b>	Data <i>goods receive</i> dan <i>goods issue</i> berhasil tersimpan.
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Aktor berada pada halaman melihat data <i>goods receive</i> yang baru.

Pengujian fitur *Goods Receive* dilakukan melalui 1 skenario, yaitu penambahan data. Pengujian berhasil dilakukan dengan data *Goods Receive* pada gambar 5.22 dan *Goods Issue* milik modul pergudangan berhasil tersimpan sesuai gambar 5.23. *Goods Issue* 94 merupakan tanggapan dari *Goods Receive* 82 / *Production Order* 82 dengan data *Production Order* pada *Goods Issue* 94.

Pp Production Goods Receive 82

[PDF](#) [Update](#) [Delete](#)

ID	82
Pp Production Order	82
Receive Date	2016-06-16 08:42:20
Storage ID	4

Pp Production Material 82							Showing 1-8 of 8 items.
							<a href="#">All</a> <a href="#">Filter</a>
#	Pp Production Request	Pp Production Order	Pp Production Goods Receive	Iwm Item Master	Planned Quantity	Actual Quantity	Availability
1	33	82	82	Frame for Deluxe Touring Bike with Black Color	100	(not set)	1
2	33	82	82	Wheels Bike	200	(not set)	1
3	33	82	82	Chain's Bike	100	(not set)	1
4	33	82	82	Handle Bar's Bike	100	(not set)	1
5	33	82	82	Seat Kit's Bike	100	(not set)	1
6	33	82	82	Pedal's Bike	200	(not set)	1
7	33	82	82	Gear's Bike	100	(not set)	1
8	33	82	82	Front and Rear Bike's Bike	100	(not set)	1

Gambar 5.22 Data Goods Receive ID 82

Good Issue 94

UpdateDelete

ID	94
Issue Name	Issue-Production-82-2016-06-19 00:00:00
Issue Date	(not set)
Sd Sale	(not set)
Pp Production Order	82
Iwm Inventory	2
Issue Type	0
Issue Status	0

Iwm Good Issue F 94

Showing 1-8 of 8 items.

All

#	Iwm Good Issue H	Iwm Inventory	Iwm Item Master	Uom	Item Demand	Qty In Inventory
1	Issue-Production-82-2016-06-19 00:00:00	Production	Touring Black Frame	Pcs	100	0
2	Issue-Production-82-2016-06-19 00:00:00	Production	Wheel	Pcs	200	0
3	Issue-Production-82-2016-06-19 00:00:00	Raw	Chain	Pcs	100	0
4	Issue-Production-82-2016-06-19 00:00:00	Raw	Handle Bar	Pcs	100	0
5	Issue-Production-82-2016-06-19 00:00:00	Raw	Seat Kit	Pcs	100	0
6	Issue-Production-82-2016-06-19 00:00:00	Raw	Pedal	Pcs	200	0
7	Issue-Production-82-2016-06-19 00:00:00	Raw	Gear	Pcs	100	0
8	Issue-Production-82-2016-06-19 00:00:00	Raw	Brake	Pcs	100	0

Gambar 5.23 Data *Goods Issue* ID 94

5.2.7. Pengujian Fitur Menambah *Goods Transfer*

Pada subbab ini akan dijelaskan mengenai hasil pengujian fitur mengelola data *Goods Transfer* yang meliputi skenario menambah data.

ID	UJ-UC-007
Referensi Kasus Penggunaan	UC-007
Nama	Pengujian Fitur Menambahkan <i>Goods Transfer</i>
Tujuan Pengujian	Menguji kemampuan aplikasi menambah <i>Goods Transfer</i>
Skenario 1	Pengguna menambah <i>Goods Transfer</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Production Order</i>

<b>Data Uji</b>	Data <i>Production Order</i> 82
<b>Langkah Pengujian</b>	Aktor masuk ke halaman daftar data <i>Production Order</i> , memilih salah satu data uji, memilih pilihan <i>Goods Transfer</i> pada data uji tersebut, memasukkan data pada form, dan menyimpan data <i>Goods Transfer</i> .
<b>Hasil Yang Diharapkan</b>	Data <i>Goods Transfer</i> yang baru berhasil tersimpan
<b>Hasil Yang Didapat</b>	Data yang dimasukkan berhasil tersimpan di basis data
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Aktor berada pada halaman melihat data <i>Goods Transfer</i> yang baru.

Pengujian fitur *Goods Transfer* dilakukan melalui 1 skenario, yaitu penambahan data. Pengujian berhasil dilakukan dengan melakukan *insert* data *Goods Transfer 1* untuk *Production Order 82* sehingga sistem modul pergudangan menghasilkan *Goods Receipt 1* sesuai pada gambar 5.24 dan gambar 5.25.

### Pp Production Goods Transfer 1


[Update](#)

[Delete](#)

ID	1
Pp Production Order	82
Iwm Item Master	Deluxe Touring Bike with Black Main Color
Quantity	100
Storage ID	2
Transfer Date	2016-06-30 08:09:30

**Gambar 5.24 Data *Goods Transfer 1***

Good Receipt 1

Update

Delete

ID	1
Receipt Name	Production-82-2016-06-30 08:09:30
Gr Document Date	(not set)
Pur Po Header	(not set)
Supplier	(not set)
Pp Production Order	82
Iwm Inventory	Finished Good
Pp Production Order	(not set)
Total Batch	(not set)
Receipt Status	0

Iwm Good Receipt F 1

Showing 1-1 of 1 item

#	Iwm Good Receipt H	Iwm Item Master	Uom	Additional Material	Item Qty	Defect	Defect Percent	Qty in Inventory
1	Production-82-2016-06-30 08:09:30	Black Deluxe Touring Bike	Pcs	(not set)	100	0	0	(not set)

Gambar 5.25 Data Goods Receipt 1

5.2.8. Pengujian Fitur *Generating Scheduling*

Pada subbab ini akan dijelaskan mengenai hasil pengujian fitur *Generating Scheduling* yang meliputi skenario konversi *Requets for Production* menjadi *Production Order*.

ID	UJ-UC-008
Referensi Kasus Penggunaan	UC-008
Nama	Pengujian Fitur <i>Generating Scheduling</i>
Tujuan Pengujian	Menguji kemampuan aplikasi melakukan konversi <i>Requist for Production</i> , yaitu melakukan penjadwalan produksi, penyusunan operasi-operasi dan pemilihan sumberdaya produksi..
Skenario 1	<i>Pengguna melakukan konversi data Request for Production</i>
Kondisi Awal	Aktor berada pada halaman daftar <i>Request for Production</i> .
Data Uji	1. Pengujian waktu produksi : <i>Frame for Deluxe Touring Bike with Red Color</i> sebanyak 100 buah

<b>Langkah Pengujian</b>	Aktor masuk ke halaman daftar data <i>Request for Production</i> , memilih pilihan <i>Convert</i> pada salah satu data, dan sistem menampilkan data <i>Production Order</i> hasil konversi.
<b>Hasil Yang Diharapkan</b>	Sistem menghasilkan jadwal dan memilih sumberdaya produksi secara tepat sesuai data <i>shift</i> pegawai dan data mesin.
<b>Hasil Yang Didapat</b>	Data hasil konversi berhasil tersimpan di basis data.
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Aktor berada pada halaman melihat data <i>Production Order</i> .

Pengujian waktu penjadwalan bertujuan untuk mengetahui tingkat optimasi waktu penjadwalan yang dihasilkan oleh Algoritma Genetika pada tiap iterasi. Pengujian dilakukan dengan melakukan konversi *Request for Production* menjadi *Production Order*. Data *Request for Production* yang akan dikonversi adalah *Deluxe Touring Bike with Black Main Color* sejumlah 100 buah dengan data *Routings* seperti pada gambar 5.26. Waktu setiap operasi pada *Routings* bervariasi sesuai dengan data yang telah ditentukan pada gambar 5.27 dan gambar 5.28.

ID		12
Iwm Item Master		Deluxe Touring Bike with Black Main Color



















Pp Routing Wc Op 12		Showing 1-8 of 8 items	
		All	
#	Pp Routing	Pp Operation	
1	12	Material Staging	
2	12	Frame Manufacturing : Frame Welding	
3	12	Frame Manufacturing : Frame Painting	
4	12	Attach seat to frame	
5	12	Attach brakes	
6	12	Test bike	
7	12	Pack Bike (Individually Packed)	
8	12	Move to Storage	

Gambar 5.26 Data *Routings : Deluxe Touring Bike with Black Main Color*

#	Pp Operation	Hrm Employee	Process Time In Minutes	
			<input type="text"/>	
1	Attach seat to frame	E017	2	
2	Attach seat to frame	E018	1	
3	Material Staging	E103	1	
4	Material Staging	E104	1	
5	Attach brakes	E105	2	
6	Attach brakes	E106	2	
7	Pack Bike (Individually Packed)	E109	2	
8	Pack Bike (Individually Packed)	E110	1	
9	Move to Storage	E111	1	
10	Move to Storage	E112	1	
11	Attach seat to frame	E117	2	
12	Attach seat to frame	E118	1	

Gambar 5.27 Data Sumberdaya Manusia



#	Pp Operation	Am Asset Active	Hrm Employee	Process Time In Minutes	
				<input type="text"/>	
1	Test bike	Testing Machine 1	E107	2	  
2	Test bike	Testing Machine 2	E108	3	  
3	Frame Manufacturing : Frame Welding	Welding Machine 1	E113	2	  
4	Frame Manufacturing : Frame Welding	Welding Machine 2	E114	2	  
5	Frame Manufacturing : Frame Painting	Spray Painting Machine 1	E115	2	  
6	Frame Manufacturing : Frame Painting	Spray Painting Machine 2	E116	3	  

Gambar 5.28 Data Sumberdaya Mesin

Tabel 5.4 Optimasi Penjadwalan Per Iterasi

Iterasi Ke	Waktu Mulai	Waktu Selesai	Durasi	<i>Fitness Value</i>	Peningkatan Optimalisasi
1	00:00:00	03:36:00	216 menit	0.00462	-
2	00:00:00	03:16:00	196 menit	0.00510	10%
3	00:00:00	03:16:00	196 menit	0.00510	0%
4	00:00:00	03:14:00	194 menit	0.00515	1%
5	00:00:00	03:10:00	190 menit	0.00526	2%
6	00:00:00	03:05:00	185 menit	0.00540	3%

Iterasi Ke	Waktu Mulai	Waktu Selesai	Durasi	<i>Fitness Value</i>	Peningkatan Optimalisasi
7	00:00:00	03:05:00	185 menit	0.00540	0%
8	00:00:00	03:05:00	185 menit	0.00540	0%
9	00:00:00	03:06:00	186 menit	0.00537	-0.5%
10	00:00:00	03:03:00	183 menit	0.00546	2%

Berdasarkan hasil pengujian pada tabel 5.4 secara umum terjadi peningkatan optimasi durasi penjadwalan. Tingkat optimasi terjadi secara bervariasi dengan rata-rata 1,94 % per iterasinya.

### 5.3. Pengujian Fitur Basis Data Terdistribusi

Pengujian Basis Data Terdistribusi (BDT) dilakukan untuk menguji 2 aspek utama dalam konsep BDT, yaitu replikasi dan *high availability*.

Sistem BDT menggunakan 4 server fisik. Dengan rincian sebagai berikut:

1. *Server aplikasi*, adalah server yang di dalamnya terdapat *web server* beserta file aplikasi ERP.
2. *Management Node*, adalah *server* yang berfungsi sebagai pusat pengaturan sistem BDT.
3. *Data Node*, adalah 2 buah server penyimpanan data.

#### 5.3.1. Replikasi

Replikasi adalah penyalinan *table-table* pada basis data ke beberapa *node* fisik yang tersebar. Operasi basis data yang terjadi pada sistem akan didistribusikan ke seluruh *node data* yang terlibat. Sehingga setiap *node data* memiliki struktur *table*

dan data yang identik. Berikut langkah-langkah pengujian mekanisme replikasi :

1. Memastikan seluruh *server* dan sistem BDT dalam keadaan *online*.
2. Melakukan penambahan, perubahan, dan penghapusan data melalui aplikasi ERP yang ditunjukkan pada Gambar 5.29.

### Warehouse 3

ID	3
Company	EZTENANT BIKE INDONESIA
Warehouse Name	EZERP Testing BDT

**Gambar 5.29 Pengujian fitur Replikasi pada sistem**

3. Melakukan pengecekan terhadap hasil penambahan, perubahan, dan penghapusan data pada seluruh *data node* yang ditampilkan pada Gambar 5.30 dan Gambar 5.31.

```

master@master-Aspire-M3970: ~
mysql> use test
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from iwn_warehouse where 1;
+-----+
| id | company_id | ws_name           | created_at           | updated_at           |
+-----+
| 3 | 1 | EZERP Testing BDT | NULL                 | NULL                 |
| 1 | 1 | Maju              | 2016-03-22 21:08:14 | 2016-03-22 21:08:14 |
| 2 | 1 | EZERP BIKE INDONESIA 2 | NULL                 | NULL                 |
+-----+
3 rows in set (0.01 sec)

mysql>

```

**Gambar 5.30 Pengujian Fitur Replikasi pada Database Server 1**

```

root@node2-Aspire-M3970: /home/node2

mysql> use test;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from iwm_warehouse where 1;
+-----+-----+-----+-----+-----+
| id | company_id | ws_name | created_at | updated_at |
+-----+-----+-----+-----+-----+
| 3 | 1 | EZERP Testing BDT | NULL | NULL |
| 1 | 1 | Maju | 2016-03-22 21:08:14 | 2016-03-22 21:08:14 |
| 2 | 1 | EZERP BIKE INDONESIA 2 | NULL | NULL |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>

```

**Gambar 5.31 Pengujian Fitur Replikasi pada Database Server 2**

### 5.3.2. *High Availability*

Aspek lain dari BDT adalah *high availability* adalah kemampuan sistem basis data untuk tetap berjalan normal pada kondisi sebuah *node data* tidak aktif. Operasi basis data yang terjadi akan dieksekusi pada *node data* yang aktif. Di lain sisi, *management node* akan menciptakan *log* operasi-operasi yang terjadi. Sehingga operasi-operasi basis data dapat dieksekusi pada *node data* yang kembali aktif. Berikut langkah-langkah pengujian aspek *high availability* :

1. Mematikan *server* aplikasi, *management server* dan sistem BDT pada salah satu *node*. Ditampilkan pada Gambar 5.32.
2. Melakukan penambahan, perubahan, dan penghapusan data melalui aplikasi yang ditunjukkan pada Gambar 5.33.

## Warehouse 3

ID	3
Company	EZTENANT BIKE INDONESIA
Warehouse Name	EZERP Testing BDT HA

**Gambar 5.32 Pengujian Fitur High-Availability pada Sistem**

3. Melakukan pengecekan terhadap hasil pemrosesan data pada *node* yang masih aktif yang ditampilkan pada Gambar 5.34.

```
root@node2-Aspire-M3970:/home/node2
id=3 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34, Nodegroup: 0, *)
[ndb_mgnd(MGM)] 1 node(s)
id=1 @10.151.64.181 (mysql-5.1.73 ndb-7.1.34)
[mysqld(AP1)] 2 node(s)
id=4 @10.151.64.182 (mysql-5.1.73 ndb-7.1.34)
id=5 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34)
ndb_mgm> show
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=2 (not connected, accepting connect from 10.151.64.182)
id=3 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34, Nodegroup: 0, *)
[ndb_mgnd(MGM)] 1 node(s)
id=1 @10.151.64.181 (mysql-5.1.73 ndb-7.1.34)
[mysqld(AP1)] 2 node(s)
id=4 (not connected, accepting connect from 10.151.64.182)
id=5 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34)
ndb_mgm> █

node2@node2-Aspire-M3970:~
[sudo] password for node2:
[sudo] init.d/mysql stop
... *
Shutting down MySQL
node2@node2-Aspire-M3970:~$ █
```

Gambar 5.33 Pengujian Fitur High-Availability pada Database Server 2

```
root@master-Aspire-M3970:/home/master
[mysql@ap1] 2 node(s)
id=4 @10.151.64.182 (mysql-5.1.73 ndb-7.1.34)
id=5 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34)
ndb_mgma> Node 2: Started (version 7.1.34)
Node 2: Node shutdown initiated.
show
Cluster configuration
[ndb@ndb] 2 node(s)
id=2 (not connected, accepting connect from 10.151.64.182)
id=3 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34, Nodegroup: 0, *)
[ndb_mgmd@ncn] 1 node(s)
id=1 @10.151.64.181 (mysql-5.1.73 ndb-7.1.34)
[mysql@ap1] 2 node(s)
id=4 (not connected, accepting connect from 10.151.64.182)
id=5 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34)
ndb_mgma>

master@master-Aspire-M3970: ~
| tax | | |
| uom | | |
| user | | |
+-----+
228 rows in set (0.01 sec)

mysql> select * from ltm_warehouse where 1;
+-----+-----+-----+-----+
| id | company_id | ws_name | created_at | updated_at |
+-----+-----+-----+-----+
| 3 | | 1 | EZERP Testing BDT HA | NULL | NULL |
| 1 | | 1 | MaJu | 2016-03-22 21:08:14 | 2016-03-22 21:08:14 |
| 2 | | 1 | EZERP BIKE INDONESIA 2 | NULL | NULL |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

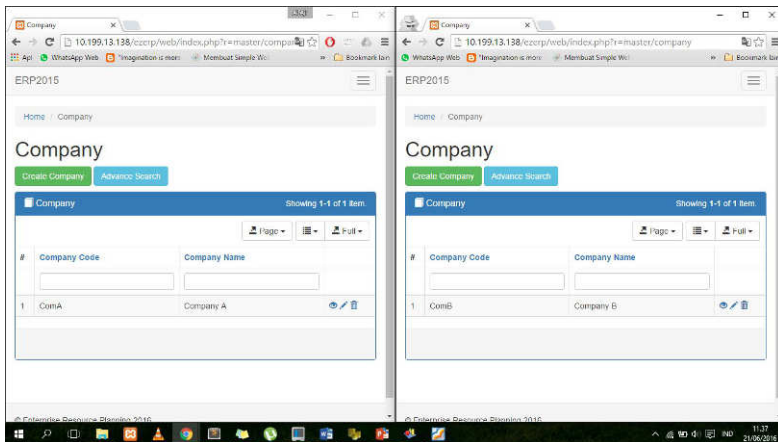
Gambar 5.34 Pengujian Fitur High-Availability pada Database Server 1

### 5.3.2.1. Pengujian Fitur Multitenancy

Pengujian ini terdiri dari pengujian mengelola *multitenancy*. Pada pengujian *multitenancy*, *tenant* baru yang akan mendaftarkan *tenant*-nya untuk dapat mengakses sistem memasukkan data berupa nama basis data yang akan dibuat. Apabila basis data berhasil dibuat, maka *tenant* dapat mulai mengakses dengan pengguna awal sebagai administrator. Hasil pengujian *Multitenancy* digambarkan pada gambar 5.35.

ID	UJ-MTC
Referensi	Multitenancy
Nama	Mengelola Multitenancy
Tujuan Pengujian	Menguji kemampuan sistem dalam mengelola Multitenancy
Skenario 1	<b>Pengguna mengakses Tenant</b>
Kondisi Awal	Pengguna berada pada halaman <i>login</i>
Data Uji	Data <i>Tenant</i>
Langkah Pengujian	Pengguna memasukkan data uji ke form yang tersedia dan memilih tombol login
Hasil Yang Diharapkan	Pengguna berada di halaman utama dengan data <i>tenant</i> -nya
Hasil Yang Didapat	Pengguna berada di halaman utama dengan data <i>tenant</i> -nya
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman utama





**Gambar 5.35 Hasil Pengujian Multitenancy**

### 5.3.2.2. Pengujian Fitur Mengelola Role Based Access Control (RBAC)

Pengujian ini terdiri dari pengujian mengelola *Role Based Access Control (RBAC)*.

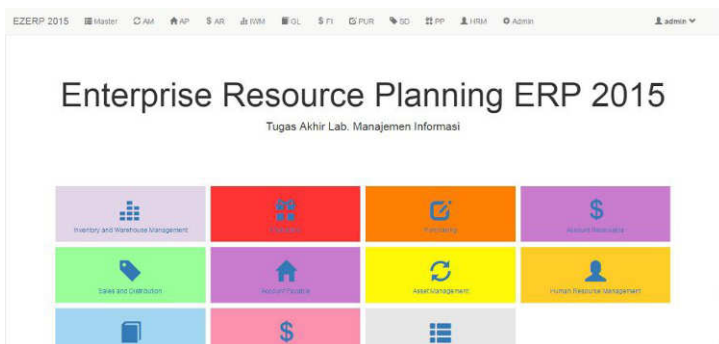
ID	UJ-UC-RBAC
Nama	Pengujian mengelola <i>Role Based Access Control (RBAC)</i> .
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data <i>Role Based Access Control (RBAC)</i> .
Skenario 1	<b>Pengguna membuat user baru.</b>
Kondisi Awal	Pengguna berada pada halaman dashboard admin.
Data Uji	Inputan data <i>user</i> baru:
Langkah Pengujian	Pengguna memilih kegiatan <i>Add New User</i> , lalu memasukkan data inputan dan menekan tombol “ <i>Add User</i> ”.
Hasil Yang Diharapkan	<i>User</i> yang baru terdapat pada halaman daftar <i>user</i> .
Hasil Yang Didapat	<i>User</i> yang baru terdapat pada halaman daftar <i>user</i> .

Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman halaman daftar <i>user</i> dengan tambahan data <i>user</i> yang baru.
<b>Skenario 2</b>	<b><i>Pengguna menyunting data user tertentu.</i></b>
Kondisi Awal	Pengguna berada pada halaman daftar <i>user</i> .
Data Uji	Perubahan data <i>user</i> :
Langkah Pengujian	Pengguna memilih kegiatan menyunting <i>user</i> tertentu, lalu melakukan perubahan data dan menekan tombol “ <i>Update</i> ”.
Hasil Yang Diharapkan	<i>User</i> yang disunting mengalami perubahan data sesuai inputan.
Hasil Yang Didapat	<i>User</i> yang disunting mengalami perubahan data sesuai inputan.
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman daftar <i>user</i> .
<b>Skenario 3</b>	<b><i>Pengguna menghapus user tertentu.</i></b>
Kondisi Awal	Pengguna berada pada halaman daftar <i>user</i> .
Data Uji	-
Langkah Pengujian	Pengguna memilih kegiatan menghapus kegiatan tertentu, lalu menekan tombol “ <i>Yes</i> ” untuk konfirmasi penghapusan.
Hasil Yang Diharapkan	<i>User</i> yang baru dihapus tidak tampil pada daftar <i>user</i> .
Hasil Yang Didapat	<i>User</i> yang baru dihapus tidak tampil pada daftar <i>user</i> .
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman daftar <i>user</i> .

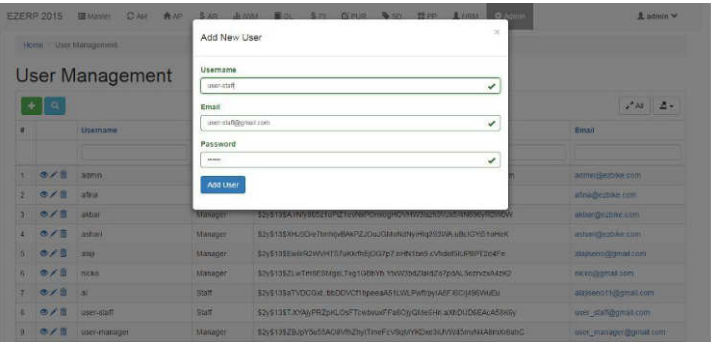
Gambar 5.36 menunjukkan proses login oleh admin. Gambar 5.37 menunjukkan hasil setelah proses login oleh admin. Gambar 5.38 menunjukkan proses penginputan data *user* untuk skenario 1. Kemudian pada Gambar 5.39 menunjukkan proses penyuntingan data *user* untuk skenario 2. Sedangkan pada

Gambar 5.40 dapat dilihat proses penghapusan *user* untuk skenario 3. Dengan melihat hasil pengujian pada ketiga skenario diatas, bisa disimpulkan bahwa Kasus Penggunaan UC.RBAC telah bekerja dengan baik seperti yang diharapkan.

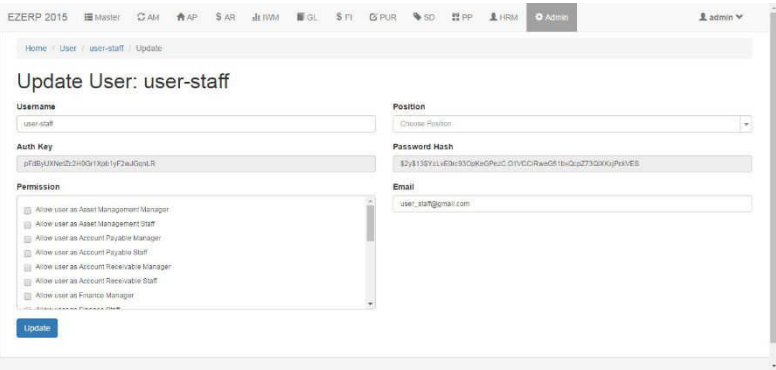
**Gambar 5.36 Proses login admin**



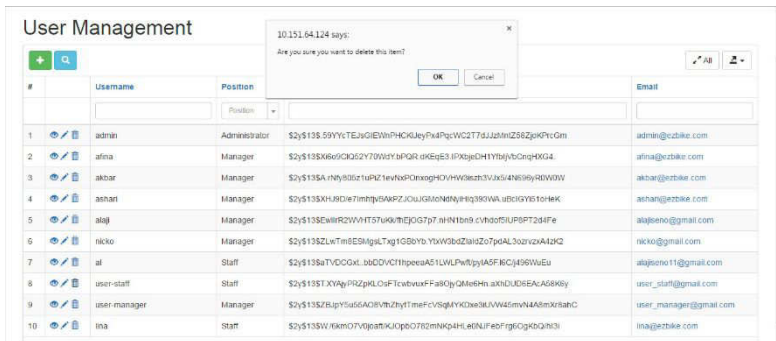
**Gambar 5.37 Tampilan awal setelah login berhasil dilakukan**



Gambar 5.38 Proses admin menambahkan user baru



Gambar 5.39 Proses menyunting data user oleh admin



**Gambar 5.40** Proses penghapusan user oleh admin

#### 5.4. Evaluasi Pengujian Fungsionalitas

Rangkuman mengenai hasil pengujian fungsionalitas dapat dilihat pada Tabel 5.11. Berdasarkan data pada tabel tersebut, semua skenario pengujian berhasil dan program berjalan dengan baik. Sehingga bisa ditarik kesimpulan bahwa fungsionalitas dari aplikasi telah dapat bekerja sesuai dengan yang diharapkan.

**Tabel 5.5** Rangkuman Hasil Pengujian

ID	Nama	Hasil
UJ-UC-001	Mengelola <i>Routing</i>	Berhasil
UJ-UC-002	Mengelola Operasi	Berhasil
UJ-UC-003	Mengelola <i>Bill of Material</i>	Berhasil
UJ-UC-004	Mengelola <i>Request for Production</i>	Berhasil
UJ-UC-005	Melakukan <i>Confirmation</i>	Berhasil
UJ-UC-006	Menambahkan <i>Goods Receive</i>	Berhasil
UJ-UC-007	Menambahkan <i>Goods Transfer</i>	Berhasil
UJ-UC-008	<i>Generating Scheduling</i>	Berhasil
UJ-UC-RBAC	Pengujian mengelola <i>Role Based Access Control (RBAC)</i>	Berhasil
UJ-MTC	Mengelola <i>Multitenancy</i>	Berhasil

*[Halaman ini sengaja dikosongkan]*

## LAMPIRAN A - RINCIAN PHYSICAL DATA MODEL

### A.1 Daftar *Table* pada PDM

Barang diterima	
PK	<u>Id_barang</u> integer
FK	Id_item integer
FK	Id_receive integer
	jumlah integer
	created_at timestamp
	updated_at timestamp

**Gambar A.1** *Table* Barang diterima

Bill of Material	
PK	<u>Id_bom</u> integer
FK	Id_item integer
	Deskripsi text
	created_at timestamp
	updated_at timestamp

**Gambar A.2** *Table* Bill of Material

BOM disusun oleh	
PK,FK1	<u>Id_bom</u> integer
PK,FK2	<u>Id_item</u> integer
	Jumlah integer created_at timestamps updated_at timestamp

Gambar A.3 Table Bom disusun oleh

Goods Transfer	
PK	<u>Id_transfer</u> integer
FK	id_item integer
FK	Id_order integer
	Jumlah integer created_at timestamp updated_at timestamp

Gambar A.4 Table Goods Transfer



<i>Goods Receive</i>	
<b>PK</b>	<b><u>Id_receive</u> integer</b>
<b>FK</b>	Id_order integer Tanggal_receive date created_at timestamps updated_at timestamps

**Gambar A.5 Table Goods Receive**

Item	
<b>PK</b>	<b><u>Id_item</u> integer</b>
	Nama_item varchar (100) Description varchar (100) created_at timestamps updated_at timestamps

**Gambar A.6 Table Item**

Job Order	
PK	<u>Id_job_order</u> integer
FK	Id_order integer
FK	Id_operasi integer
FK	Id_pegawai integer
FK	Id_mesin integer
	Waktu_mulai timestamps
	created_at timestamps
	updated_at timestamps

Gambar A.7 Table Job Order

Mesin	
PK	<u>Id_mesin</u> integer
	Nama_mesin varchar (100)
	created_at timestamps
	updated_at timestamps

Gambar A.8 Table Mesin

Operasi	
PK	<u>Id_operasi</u> integer
	Nama_operasi varchar (100) process_time integer type enum updated_at timestamps created_at timestamps

Gambar A.9 Table Operasi

Pegawai	
PK	<u>Id_pegawai</u> integer
	Nama_pegawai varchar (100) created_at timestamps updated_at timestamps

Gambar A.10 Table Pegawai

<i>Production Order</i>	
<b>PK</b>	<u><b>Id_order</b></u> <b>integer</b>
<b>FK</b>	Id_item <b>integer</b> Jumlah <b>integer</b> Mulai_produksi <b>timestamps</b> Selesai_produksi <b>timestamps</b> created_at <b>timestamps</b> updated_at <b>timestamps</b>

Gambar A.11 *Table Production Order*

<i>Request for Production</i>	
<b>PK</b>	<u><b>Id_request</b></u> <b>integer</b>
<b>FK</b>	Id_item <b>integer</b>
<b>FK</b>	Id_order <b>integer</b>
	Jumlah <b>integer</b>
	Tanggal_request <b>date</b>
	created_at <b>timestamps</b>
	updated_at <b>timestamps</b>

Gambar A.12 *Table Request for Production*

<i>Routings</i>	
<b>PK</b>	<u>Id_routings</u> <b>integer</b>
<b>FK</b>	Id_item <b>integer</b> created_at <b>timestamps</b> updated_at <b>timestamps</b>

Gambar A.13 *Table Routings*

<i>Routings</i> disusun oleh	
<b>PK,FK1</b>	<u>Id_routings</u> <b>integer</b>
<b>PK,FK2</b>	<u>Id_operasi</u> <b>integer</b>
	created_at <b>timestamps</b> updated_at <b>timestamps</b>

Gambar A.14 *Table Routings* disusun oleh

*[Halaman ini sengaja dikosongkan]*

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Pada bab ini akan diberikan kesimpulan yang diperoleh selama pengerjaan Tugas Akhir dan saran mengenai pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

#### **6.1. Kesimpulan**

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian perangkat lunak yang dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Hasil pengujian menunjukkan bahwa aplikasi ini dapat memenuhi kebutuhan fungsional Modul Produksi ERP sesuai dengan daftar fungsionalitas yang ada.
2. Hasil pengujian menunjukkan bahwa Algoritma Genetika dapat melakukan optimasi waktu *Scheduling* produksi sesuai dengan uji coba pada sub bab 5.2.8.
3. Aspek *multitenancy* dan *distributed database* dapat dan berhasil diimplementasikan pada Modul Produksi ERP sesuai dengan rancangan guna menyediakan tingkat ketersediaan yang baik pada aplikasi.

#### **6.2. Saran**

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang. Saran-saran ini didasarkan pada hasil perancangan, implementasi dan pengujian yang telah dilakukan.

1. Penambahan fitur yang dapat memberika laporan proses produksi pada hari, minggu, atau bulan tertentu.
2. Penambahan algoritma lain untuk melakukan *Scheduling*. Beberapa algoritma tersebut kemudian dibandingkan performa dan keandalannya untuk melihat keunggulan

dan kelebihan setiap algoritma dalam menghadapi sebuah kasus *Scheduling*.

3. Pengembangan aplikasi *mobile* Modul Produksi. Diharapkan dengan adanya aplikasi *mobile* dapat memudahkan pengguna yang memiliki mobilitas tinggi atau memiliki wilayah kerja/pabrik yang luas.



## DAFTAR PUSTAKA

- [1] Chong, F., Gianpaolo Carraro dan Roger Wolter. 2006. Multi-Tenant Data Architecture. Diambil dari: <https://msdn.microsoft.com/en-us/library/aa479086.aspx>. (1 Februari 2016)
- [2] Magal, S., dan Jeffrey Word. 2012. Integrated Business Process with ERP Systems. John Wiley & Sons: USA.
- [3] Suyanto. 2014. Artificial Intelligence: Searching, Reasoning, Planning, and Learning. Bandung: Informatika.
- [4] Fathansyah. 2015. Basis Data. Bandung: Informatika.
- [5] C. Bierwirth and D. C. Mattfeld, "Production Scheduling and Recheduling with Genetic Algorithms", *Evolutionary Computation*, 7(1), 1-17, 1999.
- [6] Kasman, Akhmad Dharma. 2013. Trik Mudah Menguasai OOP dengan PHP. Yogyakarta: Lokomedia.
- [7] Andrew. MySQL Cluster 7.4. 2015. New Features Webinar Replay. Diambil dari: <https://www.mysql.com/products/cluster/>. (10 Februari 2016)
- [8] Sharive. 2014. Proyek Membangun Website dengan Yii Framework. Yogyakarta: Lokomedia.
- [9] Kurniawan, D.V.Y., Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D. dan Dwi Sunaryono, S.Kom., M.Kom. 2013. Rancang Bangun Aplikasi Berorientasi Arsitektur Service (SOA) dengan Pendekatan Workflow pada Domain Production Planning Enterprise Resource Planning. Tugas Akhir pada ITS: tidak diterbitkan.

*[Halaman ini sengaja dikosongkan]*

## BIODATA PENULIS



Penulis, **Akbar Anugerah Syahdamar**, lahir di Malang, 18 April 1993. Penulis menempuh pendidikan sekolah dasar di SD Negeri Lembang Sari 04 Kab. Bekasi. Melanjutkan pendidikan sekolah menengah pertama di SMP Kartika IV-9 Malang dan SMA Negeri 5 Malang. Selanjutnya penulis melanjutkan pendidikan sarjana di Jurusan Teknik Informatika, Fakultas Teknologi dan Informasi, Institut Teknologi

Sepuluh Nopember Surabaya. Selama kuliah, penulis aktif menjadi *junior programmer* di beberapa *startup* di Surabaya.

Dalam menyelesaikan pendidikan S1, penulis mengambil bidang minat Manajemen Informasi(MI) dan memiliki ketertarikan di bidang *Web dan Mobile Application Development*. Penulis dapat dihubungi melalui email: [akbar.syahdamar@gmail.com](mailto:akbar.syahdamar@gmail.com)